DU[illegible]SRARY
NAVAL P[illegible]UATE SCHOOL
MONTEREY, CALIFORNIA 93943

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A MICROCOMPUTER TUTORIAL
ON SPREADSHEETS AND DATABASES WITH
A SIMULATED BUDGET PREPARATION

by

Sherry T. Cowen III

March 1985

Thesis Advisor: Frank E. Royer

Approved for public release; distribution is unlimited

T222024

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)*: A Microcomputer Tutorial on Spreadsheets and Databases with a Simulated Budget Preparation	5. TYPE OF REPORT & PERIOD COVERED: Master's Thesis, March 1985 / 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Sherry T. Cowen III	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Naval Postgraduate School, Monterey, California 93943	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Naval Postgraduate School, Monterey, California 93943	12. REPORT DATE: March 1985 / 13. NUMBER OF PAGES: 115
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*	15. SECURITY CLASS. *(of this report)*: UNCLASSIFIED / 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution is unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

tutorial, microcomputer, budget simulation, computer aided learning, tutorial development, spreadsheet tutorial, database tutorial

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

The objective of this thesis is to illustrate to financial managers in the Practical Comptrollership Course (PCC) some of the potential for microcomputers in budget preparation and execution. This will be accomplished through the use of a tutorial on elctronic spreadsheets and databases, and a simulated budget generated using an electronic spreadsheet. The background of microcomputer implementation into the federal government and commercial industry and the problems encountered in this (CONTINUED)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

 SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

ABSTRACT (Continued)

implementation are presented. The theory of tutorial development, along with a methodology which uses a layered procedure is discussed and used to develop the tutorial which resulted from this thesis. The tutorial manual is enclosed as Appendix A and the program is enclosed as Appendix B. It is recommended that this tutorial be included as a requirement for all PCC students.

Approved for public release; distribution is unlimited.

A Microcomputer Tutorial on Spreadsheets and Databases with a Simulated Budget Preparation

by

Sherry T. Cowen III
Lieutenant Commander, United States Navy
B.S.E.E., Marquette University, 1975

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
March 1985

C75-759
c.1

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943

ABSTRACT

The objective of this thesis is to illustrate to financial managers in the Practical Comptrollership Course (PCC) some of the potential for microcomputers in budget preparation and execution. This will be accomplished through the use of a tutorial on electronic spreadsheets and databases, and a simulated budget generated using an electronic spreadsheet. The background of microcomputer implementation into the federal government and commercial industry and the problems encountered in this implementation are presented. The theory of tutorial development, along with a methodology which uses a layered procedure is discussed and used to develop the tutorial which resulted from this thesis. The tutorial manual is enclosed as Appendix A and the program is enclosed as Appendix B. It is recommended that this tutorial be included as a requirement for all PCC students.

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943

TABLE OF CONTENTS

I. INTRODUCTION 9
A. NECESSITY FOR TRAINING 9
B. PURPOSE OF RESEARCH 10
C. SCOPE OF RESEARCH 10
D. THESIS ORGANIZATION 11

II. BACKGROUND AND PURPOSE 12
A. GROWTH OF MICROCOMPUTER USAGE IN INDUSTRY . . 12
B. GROWTH OF MICROCOMPUTER USAGE IN THE DEPARTMENT OF DEFENSE 15
C. PURPOSE 20
D. SUMMARY 22

III. CONSIDERATIONS IN DESIGNING TUTORIALS 23
A. BACKGROUND 23
B. METHODOLOGY 26
1. Definition 27
2. Local Structure 28
3. Presentation 31
4. Site Implementation 33
C. SUMMARY 33

IV. PROJECT DESIGN 35
A. INTRODUCTION 35
B. DISCUSSION 35
C. DEFINITION 36
1. Goal One 36
2. Goal Two 37
3. Goal Three 38
4. Goal Four 40

D. LOCAL STRUCTURE 40
E. PRESENTATION 41
F. SITE IMPLEMENTATION 43
G. SUMMARY 49

V. CONCLUSIONS 51
A. INTRODUCTION 51
B. DISCUSSION 51

APPENDIX A: TUTORIAL GUIDE 53
A. GETTING STARTED 53
1. Specifications 53
2. Setting Up 53
3. Power-up 55
4. Initializing the Machine 55
5. Getting Out of the Tutorial 56
B. MAKING A SELECTION 58
C. ELECTRONIC SPREADSHEET USING LOTUS 1-2-3 . . . 59
1. Introduction 59
2. Display Screen 60
3. Tutorial 63
4. Completing a Spreadsheet 63
D. BUDGET ANALYSIS 70
E. DATABASE TUTORIAL INTRODUCTION 75
1. Purpose of a Database 75
2. Tutorial 76
F. LEAVING THE TUTORIAL 91

APPENDIX B: TUTORIAL SOFTWARE 93

LIST OF REFERENCES 113

BIBLIOGRAPHY 114

INITIAL DISTRIBUTION LIST 115

LIST OF TABLES

I Data for Student File 82

LIST OF FIGURES

4.1 Program Modules 47
A.1 Inserting floppy disk Reproduced from Compaq Manual 56
A.2 Main Menu 56
A.3 Spreadsheet Screen 60
A.4 Control Panel 61
A.5 Worksheet 62
A.6 Completed Spreadsheet 74
A.7 Screen to Define a File Structure 80
A.8 File Structure 81
A.9 System Ready to Receive Input to Records 83

I. INTRODUCTION

A. NECESSITY FOR TRAINING

With the cost of microcomputers steadly decreasing, at the same time as their processing capability is increasing, automated budgeting and budget execution software is being implemented or encouraged at many commands. Commander, Naval Logistics Command, Pacific Fleet, for example, has designed and distributed to all its' subordinate commands a microcomputer based, budget preparation, submission and execution application software package. This package is designed using a commercially prepared spreadsheet software package. It allows the user to develop their budget on the microcomputer in the format required by COMNAVLOGPAC. To submit their budget, a command merely makes a a copy of the budget spreadsheets on a separate disk and forwards it to COMNAVLOGPAC. As the budget worksheet disks are received, COMNAVLOGPAC updates their total budget from the disks. As the budget year progresses commands update the COMNAVLOGPAC data by sending an updated disk with their current status recorded on it. This package will be used for the submission of the next annual budget by the subordinate commands.

The financial manager that must use microcomputers and the associated software to implement the commands own automated budgeting system, or one distributed from a senior command, has little training available within the Department of the Navy that prepares the manager to use modern hardware and software. The requirement for a financial manager to use and understand microcomputers and specific application packages has not existed in the past. Now the financial

manager finds that the requirement for this knowledge is becoming a necessity as the proliferation of microcomputers continues throughout the federal government [Ref. 1: p. 23].

B. PURPOSE OF RESEARCH

Given this lack of formal training within the Department of the Navy, the primary objective of this thesis is to develop a tutorial explaining the use of a typical commercially available spreadsheet and database, and the use of such a spreadsheet in the preparation of a command budget. The tutorial will be used by the students of the Navy Practical Comptrollership Course as an introduction to microcomputers, spreadsheet and database software packages and their use in budget preparation and execution.

C. SCOPE OF RESEARCH

This thesis will examine the introduction and growth of the microcomputer industry and their implementation into the Department of Defense (DOD) and commercial industry. The regulations within the the Department of the Navy (DON) which govern the procurement of microcomputers will be reviewed.

The theory behind developing computer aided learning (CAL) material will be discussed and a particular methodology will be presented and used to implement a microcomputer based tutorial. The tutorial will be directed towards financial managers in the Navy who are attending the Practical Comptrollership Course(PCC) at the Naval Postgraduate School, Monterey, Ca.

There are thirteen major claimants, all with different budget input and database requirements. Therefore the spreadsheets, databases and budgets will be generic and not directed towards a specific command.

D. THESIS ORGANIZATION

Chapter One has discussed the necessity for training, the purpose of the research and the scope of the research. Chapter Two will discuss the growth of microcomputer usage in industry and the federal government. It will also discuss the void in training in microcomputers for the Navy's financial managers and the need for this void to be filled. A project is proposed to develop a microcomputer based tutorial to fill the void. Chapter Three will discuss theory behind designing computer based courseware and a methodology to be used in developing this project. Chapter Four will discuss how the methodology presented in Chapter Three was applied to the project. Each step of the project is broken down into its separate components and expanded to result in the final product. The hardware and software required to use the tutorial which resulted from this project are discussed in Chapter Five. Chapter Six recommends how the tutorial should be used and monitored in the future. There are two appendices. Appendix A contains the tutorial manual to be used in conjunction with the course. The software listing is located in Appendix B.

II. BACKGROUND AND PURPOSE

A. GROWTH OF MICROCOMPUTER USAGE IN INDUSTRY

Since the first introduction of non-tactical computers into the federal government by the Census Bureau in 1952, the number of computers in use by the various agencies has continued grow. This growth was slowed by the enactment of the Brooks Act of 1965. In the early 1960's Congressman Brooks saw the continued growth of computers in the government beginning to spread in an uncontrolled manner, with few restrictions. Most systems were acquired with sole source contracts, without attempting to receive competitive bids from the various manufacturers. The Congressman felt that this procurement procedure was raising the governments costs in computer procurement. Therefore the Brooks Act of 1965 was introduced and passed to ensure the federal government procurement of computers was as economic and efficient as possible. It did not stop the growth in use of computers in the federal government but, rather, served to slow the growth somewhat.

In recent years the advances in technology has reduced the size and cost of computers, giving birth to the microcomputer. The original computers required rooms of cabinets filled with the state-of-the-art tubes. These systems were programmed in machine language code which was unintelligible by anyone other then the trained computer scientists. Programming the systems was a long process which involved the setting of a series of switches in the on or off position. The systems were designed to work on one process at a time. With the introduction of transistors and integrated circuits the size and power requirements were

reduced to a small percentage of what they had previously been. The processing capabilities of microcomputers increased in speed and also included the capability to process more than one program or user at the same time. With the advent of very large scale integration circuits (VLSI), which resulted in the development of the processor-on-a-chip, came the birth of the microcomputer. The capabilities of the microcomputer increased from the single user low level language system to the multiuser, 4th generation language systems.

When microcomputers were first introduced on the market they were not much more than a toy for the wealthy due to the high cost, or a challenge for the electronic hobbyest due to the small internal memory size and lack of commercially available software. They were normally purchased as part of an electronic hobby kit or correspondence training course. The user was required to assemble, test, and program the computer. When assembled it would solve problems which would be considered trivial to the user of a modern microcomputer.

Software written for the first microcomputers was not portable; it could not be used on microcomputers manufactured by different companies. Most of the software was produced for a particular equipment configuration. The software which was available commercially had programming faults (bugs), and was not sophisticated enough to be used in commercial or office applications. These problems with software kept the business world from exploiting the potential of early microcomputers.

In addition to the problems of software availability, the first microcomputers were not reliable enough to be considered for use in a business or office environment. The quality assurance problems with these first microcomputers is documented by the major recall by Apple computers of one of their first microcomputer lines due to design problems.

In 1980 the advances in technology and reliability of the hardware and the availability of dependable, commercially packaged, portable software were at a level acceptable to the business world. In 1981 International Business Machine (IBM), introduced its' first microcomputer. Now with the worlds largest computer company manufacturing microcomputers, the business world began to look at them as more than just a toy. Software development companies started to produce a wide variety of software packages in larger quantities, and of better quality. Although there were no required or accepted standards for the microcomputers, many manufacturers attempted to develop systems that were compatible with the IBM microcomputer. This unofficial standardization provided the the necessary ingredient to allow the software to become portable.

The typical microcomputer system is configured with the following equipment components; a processor unit, read only memory (ROM) which contains instructions to handle routine internal functions, random access memory (RAM) which stores the data or program currently in use, secondary storage devices such as floppy and hard disk drives to store data and programs outside of the main memory, a monitor for the computer to output to the user using a non-permanent medium, and a printer to output to the user with a permanent medium. The early 1980's saw the development of microcomputers with greater processing power, increased memory size, and faster secondary storage devices. Four and eight bit systems were replaced by sixteen and thirty-two bit processors. Increasing the bit size of the processors increases the amount of data which can be handled in a step by the processor, which translates into faster operations. Sixteen and forty-eight thousand byte memories were replaced by two hundred and fifty-six thousand byte and larger memories. Cassette tapes were replaced by floppy and hard disk drives.

This greater processing power, increased memory size, and faster secondary storage devices, combined with the advances in software prompted the acceptance and increased use of the microcomputer in business.

B. GROWTH OF MICROCOMPUTER USAGE IN THE DEPARTMENT OF DEFENSE

The introduction of the microcomputer into the federal government has been slower than in the private industry. One of the reasons for this has been the procurement regulations which the agencies of the federal government must operate under. As an example, in the Navy, the purchase of equipment falls under two categories; expense and investment appropriations. Expense appropriations finance the cost of ongoing operations and maintenance (O&M,N), military pay (MPN), and family housing (FHN). The local commander can finance the daily operations of his command through O&M,N funds provided in an Expense Operating Budget (EOB), and can use O&M,N funds for operating expenses and equipment purchases up to a $3000 per system limit. Public law prohibits the procurement of any equipment which exceeds $3000 per line item from O&M,N funds. Further complicating the procurement of microcomputers for a local commander is the Navy's definition of a computer system. All components of a microcomputer, such as the processor, monitor, printer, modem, and expansion memory boards, are considered together in defining the system. The sum of the cost of the components must be less than $3000 if it is to be purchased from O&M,N funds.

Investment funds on the other hand are for major procurements and construction. Procurement covers specific items such as aircraft (APN), weapons (WPN), shipbuilding and conversion(SCN), and other items over $3000 (OPN).

Procurement of non-tactical computer systems costing over $3000 would occur under OPN funding. OPN funds are normally distributed from the headquarters commands for specific projects that have been approved through the chain of command.

Until recently, the cost of a microcomputer system had been greater than the $3000 limit for the O&M,N appropriation and could not be purchased utilizing O&M,N funds. To purchase a system would require procurement using the procedures of Secretary of the Navy Instruction (SECNAVINST) 5230.6, Delegation of Automatic Data Processing Approval Authority and Acquisition/Development Thresholds. Under this instruction microcomputer acquisition would fall into two possible categories:

1. Annual aggregate total cost for purchase or rental of automated data processing equipment (ADPE), software or services does not exceed $10,000.
2. ADPE, software or services is greater than $10,000.

If a microcomputer system costs less than $10,000 the system would be procured using OPN funds on a local small purchase contract. If a microcomputer system will cost less than $10,000 the system would be subject to Automati Data Processing (ADP) approval level thresholds. These thr shold levels are:

1. A $50,000 limit for the lowest level approval for a system which a commander desires to procure through a sole source contract.
2. A $300,000 limit for the lowest level approval for a system which a commander desires to procure through a competitive bid process.

If a computer system costs less than $50,000 the lowest level commander, such as Commander, Naval Telecommunications Command (COMNAVTELCOM), could approve procurement of the system through a sole source contract. If a computer system

will cost less than $300,000 COMNAVTELCOM could approve procurement through a competitive bid process. To receive the approval for the acquisition of ADPE a command must submit a Mission Element Need Statement (MENS), in accordance with SECNAVINST 5231.1, Major Automated Information Systems Approval Process. A MENS describes a mission deficiency, provides a study of alternative solutions to correct the deficiency, and identifies the estimated costs associated with each action to the decision maker. Submitting a MENS for approval to one of the ADP approval authorities puts the command in competition for the limited dollars the particular authority has for ADPE.

Another reason for the slow introduction of the microcomputer into the Department of Defense (DOD) was the size and age of the federal government. This old established system is made up of many old established agencies with old established practices and methods of handling data. An influx of microcomputers would have required the rethinking and redesigning of the established methods of doing business. The profit motive is not present in the government as in most commercial enterprises which will bring about change if shown a more profitable way of doing business. Cost effectiveness, which is of concern to managers in a government agency, is a motivator which is not as powerful as the profit motive. Cost effectiveness analysis is normally based upon showing more benefits from the idea or system under consideration than the costs to implement the idea or system. If all the benefits were tangible ones there would be no questions as to the cost effectiveness. But when most of the benefits are intangible ones the value of a system becomes more a question of how one weighes the benefits against the costs. There were few young new agencies being started which would not have to redesign their methods of doing business in order to integrate computers into their daily procedures.

Private industry was not hampered by all the same factors affecting the slow rate of microcomputer introduction into the federal government. The procurement procedures of commercial corporations vary between each company. Some have requirements which are as involved as those in the federal government, while others have minimal restrictions on the procurement of microcomputers. This range of procurement requirements resulted in some companies being quicker than others in introducing the microcomputer into their daily routine. There were always new companies coming into existence which did not have to overcome a corporate fear of computers that may have existed in an older company. Establishing their organization around a microcomputer system was relatively easy and a potentially profitable idea if properly accomplished.

In the last few years, the influx of microcomputers into the federal government has greatly increased. There was an 11.3% increase in spending on ADPE hardware in 1983, with the largest increase going towards the purchase of microcomputers [Ref. 2: p. 1]. In October of 1983 a contract was let to purchase 6000 Zenith Data Systems microcomputers over three years by the Air Force, Navy and Marines for productivity improvement applications. 17000 applications for the 6000 micorcomputers were received in the first eight months after announcement of the contract [Ref. 3: p. 62].

Because of this sudden growth of microcomputers in the federal government, the General Services Agency (GSA) is attempting to establish government wide initiatives to control the proliferation of microcomputers and assist agencies in making intelligent decisions regarding the procurement of microcomputers [Ref. 1: p. 44]. These initiatives include:

1. Development of precise justification procedures.
2. Establishment of review and approval procedures.

3. Establishment of an evaluation program.
4. Establishment of a buyer's guide for microcomputers.
5. Development of procurement vehicles for hardware and software.
6. Development of a nationwide maintenance contract.

The GSA Office of Information Resources Management has distributed a handbook, "The End Users Guide To Buying Small Computers". This handbook walks the non-ADP manager through the steps of planning and procurement of a microcomputer. It aids the federal employee unfamiliar with microcomputers, and reduces the uncertainty about obligations under the federal procurement regulations. The cost of many microcomputers systems configured for typical business applications has fallen below the $3000 limit, making it possible for local commanders to purchase a system costing less than $3000 without external approval. Requests and justifications must only be routed and approved within the local command. Requests get the priority in both time and dollars that the local commander desires, greatly reducing the time from inception of a new system to implementation. The GSA has opened 2 stores, one in Washington D.C. and a second in Philadelphia, which carry 16 lines of microcomputers. These stores provide free training, equipment burn-in, (running the equipment for two hours to check the system for problems), and a thirty day unlimited guarantee. The first store in Washington D.C. estimated $2.5 million in sales for the first year, but actually exceeded $8 million. [Ref. 4: p. 44].

This influx of microcomputers into the federal government will affect the financial managers of the Navy. Financial managers at commands subordinate to Commander, Naval Logistics Command Pacific, all received a microcomputer and software this past year, and have already been affected by the microcomputer expansion. The

instructions received with the package stated that all commands were to use them for their next budget submission and budget execution cycle. These commands cannot avoid directing their interest towards the operation and usage of a microcomputer.

The Navy currently offers little instruction in microcomputers. An informal survey of the January and February 1985 Practical Comptrollership Course (PCC) students indicated that more than half of the students (30 out of 59) had never used a microcomputer. Of those who used microcomputers at work one person indicated they had received the benefits of formal command training, ten were self taught, and sixteen had received the benefits of on-the-job training or a manufacturer's tutorial. Two indicated they were essentially untrained. If these results are indicative of training provided to financial managers throughout the Navy, these managers are ill-prepared to to utilize current microcomputer technology. Many of the managers received both their education and early work experience at a time prior to the wide introduction of the computer technology. If they do have experience with computers in their education or work experience it was most likely with the older, unsophisticated, batch type operating systems. This provides little value in dealing with the challenges of today's microcomputers.

C. PURPOSE

The PCC is an ideal arena to give a wide variety of the Navy's financial managers the opportunity to obtain hands-on experience with a microcomputer. The students of the PCC attend the course at the Naval Postgraduate School (NPS) in Monterey, CA, an environment which stimulates a wide variety of learning experiences during the two week duration of the

course. Students would not have the pressure of their normal daily work requirements keeping them from being able to spend time learning the new technology and how it can help them. There is also the expertise available at NPS to assist them when they encounter difficulties. An introduction to the current microcomputer technology available in in this setting would enhance a financial managers abilities to oversee the implementation of the technology at their own commands. This does not mean that their experience would make them experts in the field, but the training would allow them to better understand the new systems introduced into their commands.

The purpose of this thesis is to provide a microcomputer based tutorial for the financial managers attending the PCC. The primary objective is to write a tutorial that could be implemented on a portable microcomputer which could be carried back to the students motel room. With a tutorial guide and a short presentation in the class, the student could complete a hands-on tutorial in three to four hours during the evening. Upon completion of the tutorial the student should have gained a rudimentary expertise in operating a microcomputer.

A secondary objective of the tutorial is to provide the student with a basic understanding of various off-the-shelf software programs that may be useful in accomplishing their routine daily assignments. The student will gain an understanding of off-the-shelf spreadsheet programs and how they are similar to the manually manipulated spreadsheets they currently use. They will also gain an understanding of how an electronic spreadsheet is constructed by putting the knowledge gained in the tutorial to use in completing a worksheet. A realization of the flexibility provided by electronic spreadsheets in developing "what-if" scenarios in budget preparation and execution would be provided through

the use of a simulated budget exhibit. This section of the tutorial is designed to simulate the budget submission and adjustment phase of the budget cycle, showing how the electronic spreadsheet will quickly analyze and summarize alternative actions available to a command.

The tutorial would also provide an introduction to off-the-shelf database programs. The student would receive some basic database theory, be allowed to manipulate a small database and receive a rudimentary understanding of the power of inexpensive database software in maintaining a commands information.

D. SUMMARY

The influx of microcomputers into the DOD will soon be affecting most managers. The tradition of first implementing computer systems in the comptroller department has been broken in the case of microcomputers. The user friendliness of microcomputer software has allowed non-technical personnel in all departments to develop their own microcomputer systems at the same rate as the comptroller department. Financial managers are still being moved towards the usage of these systems faster than they are able to find training. This requires the financial manager to search for this training on their own. The Practical Comptrollership Course is an ideal location to make this training available to them, because this course is attended by a large cross-section of the Navy's financial managers.

The microcomputer training must be developed using proven methods in tutorial development which will provide an organized, step-by-step procedure. The training must also consider the potential student population which will use the tutorial. A method which will accomplish this is introduced in Chapter 3.

III. CONSIDERATIONS IN DESIGNING TUTORIALS

A. BACKGROUND

The use of computer aided learning (CAL) has been increasing since the advent of the microcomputer. Prior to the widespread availability of microcomputers, CAL was limited to large institutions, such as universities, which had mainframe computers to use in the instructional process.

The concept of CAL is not really a new one, rather it is the next logical step in individual instructional techniques. The sophistication of these self teaching techniques has progressed with along with microcomputer technology. Starting with self-paced printed material these techniques gradually encompassed the use of audio tapes, video tapes, and modern day computers. Within each of these different instructional media there have been different approaches used to present the material. Written instructional materials came in various forms, with most falling into one of three general categories. The textbook format presents material in chapters, with questions provided at the end of each chapter. The answers to the questions are not normally provided to the student. The tutorial format presents the material by topics and then presents questions to the student on the material just covered. The answers to the questions are provided to the student. The programmed text presents a topic, then asks the student a question. Depending upon whether the question is answered correctly or incorrectly, the student is directed to either new material or a review of the previous material. The audio and video tapes were usually limited to the material being presented by an instructor in a manner

similar to a classroom presentation. Once the presentation was on the recorded medium it was available for use whenever the student wanted to use it.

CAL has developed in a manner similar to that of written instructional materials. Although the medium used is a screen instead of paper, the information is transferred to the student with words and drawings on the screen in a manner similar to that of printed material. There is computer aided instructional material that is pre-structured and that which is adaptive [Ref. 5: p. 33]. Pre-structured CAL is programmed to respond to only a selected few users responses. This structure guides all students through the material in the same manner. An adaptive system is programmed to respond to a wide variety of responses. This will allow the system to guide each student through the course work based upon their responses to the questions in a way which is best for each particular student.

According to Crawford [Ref. 6: p. 1] CAL has five major advantages:

1. As opposed to traditional modes of instruction, the student is rarely able to be a totally passive receiver of information. When using a CAL program, the student is generally an active participant in the learning process.
2. CAL has the potential to be a highly individualized method of instruction. Students can proceed at their own pace and can respond to questions without the fear of embarrassing failure. In addition, students can receive important individualized feedback regarding progress being made.
3. Students can simulate experiments or explore situations which would be too dangerous, expensive, difficult or time consuming to undertake in the real world.

4. When computers are used as an aid to instruction, classroom teachers can become "facilitators" of instruction rather than lecturers. In addition, teachers generally have more free time to devote to dealing with students in a more personal and meaningful manner.
5. The computer evaluates a student's performance in a completely objective manner, and the possibility of an unconscious teacher bias is eliminated.

With these five, strong advantages in its favor, it would appear that CAL will be a major instructional tool in the near future. For this to occur, however, CAL must change its image. Currently CAL is grouped into two camps; smooth and rough [Ref. 5: p. 104]. The smooth material is normally highly effective and prepared by professionals. It is presented to the student in a sophisticated fashion. This type of software is expensive due to the time required to develop and the highly experienced programmers required to work on its development. The rough material is normally prepared quickly as a stop gap measure, by less experienced programmers. It appears less sophisticated and is normally less effective than it could be if properly developed. It is less expensive to develop due to the shorter time expended in its development, although, in the long run it will be more expensive because it will require replacement when its ineffectiveness is realized. Due to its low cost the less sophisticated CAL material has been getting wider use and has been the material by which CAL as a whole has been judged.

One way CAL can change its image is through use of a proven methodology and development strategy. One of these methods is discussed later in the methodology section of this chapter. Barnes [Ref. 7: p. 31] discusses six commandments which an author of CAL should keep in mind no matter what methodology is used.

1. 'Thou shall not run books cum disks on your computer." Reading page after page on the computer screen is not only boring but also a waste of the computer as a teaching resource. Books are more portable than a computer. A computer should give the student conversation during a teaching session.
2. "Keep the student involved." There should be some interaction between the student and the computer. This can be in the form of the student answering questions or inputting data.
3. "Timed screens shall be used sparingly if at all." This type of limitation tends to pace the lesson instead of allowing the student to advance at a comfortable pace. Not everyone reads at the same speed.
4. "Always use, but don't overuse computer tricks like graphics, blinking, and animation if available." Color can be an effective tool for emphasis, and also adds to the enjoyment of the student. Blinking can quickly lose its effectiveness and appeal if used more than a couple times.
5. "Make your games make sense." Not all subjects make sense as a game. Where math is a good subject to use in conjunction with a game, the topic of artificial respiration is not.
6. "Create a simple user interface." Unless the course subject is on teaching computer operations, the complex operations have no place in the course.

B. METHODOLOGY

The methodology used to develop a tutorial consists of four distinct layers or steps. These layers begin with the definition layer, progress through the definition of local

structures and presentation method, and end with site implementation of the tutorial. [Ref. 5: p. 10]. This layered method ensures the design is completed separately and independently from the particulars of the implementation.

1. Definition

This layer identifies the goals, objectives, and rules to be used in the development of the tutorial. The goals are made up of one or more objectives and the objectives are made up of one or more rules.

Defining the goals that the course is going to achieve is the initial step in designing CAL courseware. These goals should be stated in broad, general terms which state the results the students will achieve in using the course material. These goals will provide the general direction in which the material will proceed [Ref. 6: p. 19] and defines the depth as well as the breadth of the material to be covered. A major consideration in defining the goals is an analysis of the prospective users and the environment in which they will be using the course material.

The second step in the definition layer is to define the objectives which will steer the student into meeting the defined goals. These are narrower in scope than goals, and stated in terms of what the student will do. One or more objectives will be required to meet each goal.

The last step in the definition layer is to develop the rules or tasks which accomplish each objective. The term rule in this sense means a required syntax for a particular software package, a definition or statement about a concept [Ref. 5: p. 21]. Tasks are the steps the student must take to attain an objective.

2. Local Structure

There are two types of local structure in CAL; pre-structured software and adaptive software. CAL software which is programmed adaptively is a better system to guide a student through a course since it will tailor itself to the needs of each student. But there are some disadvantages to an adaptive system. It is more expensive to create due to the experience required to develop the program. It also requires an extensive amount of time to develop. Since a totally adaptive system would be similar to a private tutor who could respond to any response by the student the hardware required to implement a totally adaptive system, which would respond in a reasonable amount of time, is beyond the capabilities of most of todays' microcomputers. [Ref. 5: p. 33]

A pre-structured CAL system is on the opposite end of the instructional spectrum. The main advantage of a pre-structured CAL is that it is much easier and less expensive to develop. Because of this there are many of these systems of varying quality in existance. A disadvantage of pre-structured CAL is that it takes all students through the course material in the same manner without giving consideration to the individual students needs. Most courses are not totally pre-structured or totally adaptive; rather, they tend to display features of both of these structures. As computer systems become more capable and sophisticated the courses are shifting more towards adaptive systems due to the advantages they offer.

According to Godfrey and Sterling [Ref. 5: pp. 33-57], there are five basic CAL teaching strategies; drills, test, inquiry, simulation, and tutorial. While each of these strategies can be designed as either adaptive or pre-structured, their effectiveness could be limited if they are misapplied.

In a drill strategy, a question is presented to the student. The response to the question is then checked against the possible answers. If the answer is correct the student is informed and the program continues on to new material. Incorrect answers fall into two categories; anticipated and unanticipated. If the incorrect answer was anticipated the student is given feedback to correct the deficiency in the form of a similar question to stimulate the correct response or a review of the concept involved. An unanticipated incorrect response can only be assumed to be wrong and the system should give the correct response to the stulent. Drills are good for memorizing simple facts, but are not useful in developing complex or imaginative thought.

If a testing strategy is used the only building block is the question. There is no attempt to evaluate responses other than to determine whether it is correct or incorrect. This type of strategy can be used to evaluate a knowledge level gained in a previous experience or to apply the knowledge creatively to develop a new skill. Testing is applicable in determining a persons level of knowledge in a particular subject or in determining the cause of continuous error. A shortfall of the testing strategy is that a student may gain knowledge from a course which is not demonstrated in a test.

Inquiry CAL is the opposite of testing. In an inquiry system the user selects the information to be covered. The selected subject is then displayed for the student. There are no questions at the end of the session to test the knowledge level achieved by the student. The only questions utilized are involved with determining the subject the student chooses to view such as selecting a topic from a menu. The inquiry strategy works well with other local structure strategies to make a complete system.

It is also a good technique to use when the possible users will have a wide range of previous knowledge in the subject material since it allows them to choose the material they will view. It can cause some frustration, however, if the student must go through several menus to get to or out of a particular topic.

The simulation strategy is used to model a real life scenario. A true simulation CAL will have a large branching structure which will allow as many possible variations as the real life situation would allow. Since this strategy attempts to recreate real life situations there are generally no right or wrong responses. Simulations are good for reproducing real life situations which would be too expensive or dangerous to actually accomplish. Examples of simulation systems are flight simulators used to train pilots, nuclear power plant simulators used to train civilian nuclear power plant operators, and weapons training simulators used to train the weapons system operators. Simulations are also ideal for demonstrating cause and effect situations. They can, however, be difficult to design if the rules of cause and effect are detailed and involved.

The CAL which uses a pure tutorial strategy should be made as adaptive as possible. The tutorial gives the student the rules which define the topic, then shows examples of the rules in use. Finally questions are asked to determine the level of the students knowledge. Analysis of the students response is what makes a tutorial as powerful as it is and, therefore, difficult to develop. The tutorial should determine, from the students response, what type of feedback the student should receive. This could be one of various responses which include asking another question on the same or another subject, reviewing the subject matter, or going on to the next subject. A tutorial

strategy is good when utilized to train an individual in the use of the computer and when students who will use the system have various levels of backgrounds. A tutorial, however, requires a large amount of computer memory to store the data and ties the computer to one person for a long period of time.

Crawford [Ref. 6: p. 35] introduces another type of CAL he calls dialogue CAL. In dialogue CAL the student receives the rules, examples and questions from the computer as in a tutorial, but when the student begins to have difficulties the system converts to the dialogue mode where the stulent asks the system questions. These questions must then be interpreted by the system. Due to the many different ways questions can be phrased, the form of the questioning is normally limited to a set format. Due to the extensive development required, dialogue CAL should be analyzed for sufficient returns prior to expending effort and money on its development [Ref. 6: p. 36].

3. Presentation

The presentation of the instructional material refers to the method used to introduce the student to the course material. In the presentation step a developer analyzes the media to be used. He may consider providing the stulent with written material, displaying information on the computer screen, and using the keyboard as an input device.

When using CAL in a microcomputer, design of display screens is an important step. The display format should be consistent between screens. This allows the student to become familiar with the format and eliminates the need to search each screen for the location of the expected data or instructions. The screen should be divided into windows which display the data in the same format each time a new screen appears.

Graphics are used to present a picture or an idea on the screen and may supplement the use of text or amplify what is in the text. Graphics are hardware dependent and therefore greatly reduce the portability of CAL. When designing graphics, consideration must be given to whether a user will use monochrome or color monitor and whether that monitor is a high or low resolution device. A static display which is the same as a still picture and a dynamic display which is similar to a animated cartoon must also be considered. Both static and dynamic graphics can be effective in CAL. They can help hold the students interest and a good picture on a monitor is still worth a thousand words. There are some disadvantages with each. Both add a substantial amount of time to the development process and reduce the portability of the software between computer systems. In addition, dynamic graphics can cause the user to become engrossed with the action on the screen and miss the lesson being presented.

The keyboard is not the only input device available. There are also light pens, joysticks, mouse, and voice recognition devices. All of these are acceptable input devices, although voice input devices, at a reasonable price, are not reliable at this time.

When considering the method of output the computer monitor is the most convenient, but use of printed and audible output is also acceptable for some forms of communication. A printed output would be better suited for a hard copy of a terminal session while audible output works well during the active session. The advantage of monitor output is that it is fast and doesn't require paper to output to the user. A disadvantage of the monitor is that the output is normally lost once it has been moved off the screen.

4. Site Implementation

Site implementation converts the material developed by the previous three steps into a usable product for a specific system. This step takes into consideration the hardware and software to be used. It is during this phase when the designed systems are prototyped, tested and evaluated.

During prototyping of the system the designer will actually see, on the screen, the complete package that will be presented to the student. There are times when this may cause a step backward to the presentation phase to redesign the screen formats or include additional information.

When the prototype system has been completed the system should be tested by potential users or people with similar backgrounds to the potential users. These testers should give feedback to the design team giving their evaluation of the system which should include clarity, fluidness and applicability of the tutorial, and amount of training received. The system should be tested to see if it is operative, and "bullet proof". "Bullet proofing" implies that the system has sufficient fail-safe mechanisms built in to prevent the operator from causing the system to go off into infinity by pushing the wrong key. An advantage of system design being accomplished by the same organization which will implement it lies in their knowledge of the hardware limits on which it will be run. This can save an organization from having to go back to rework some of the previous three stages.

C. SUMMARY

The development of CAL as a teaching medium is a continuation of the adaptation of technology to the field of education. Development of CAL requires following a set of

procedures such as the methodology presented in this chapter to prevent the goals being achieved from possibly being altered in the implementation process. This process of top-down development requires a complete development of a layer before moving onto the next layer to ensure the later step does not alter the results of the previous layer.

The particular methodology presented in this chapter was applied to the development of the tutorial which resulted from this project. This development is described in Chapter 4.

IV. PROJECT DESIGN

A. INTRODUCTION

The tutorial which resulted from this study was designed using the methodology discussed in Chapter 3. These guidelines allowed the project to proceed in an organized manner, and resulted in a product which met the authors goals. This chapter will discuss the steps taken in designing the tutorial included as Appendices A and B.

B. DISCUSSION

There is a requirement in the design of any tutorial to look at the potential users and the environment in which they will be using the course material. The potential users of the courseware developed in this thesis are all financial managers enrolled in the Navy's Practical Comptrollership Course. Beyond the area of their employment they have various backgrounds and experience. This is especially true in their experiences with computers. In an informal survey conducted in the January and February 1985 PCC, student responses indicated that half of them had no experience with microcomputers. The remainder had experience which varied from novice all the way through expert level, with the lesser experience levels being more predominant.

It was decided early in the design phase that the course will be taken by the students to their motel rooms, to be accomplished during the evening hours or over a weekend. There are two reasons behind this decision. First the length of the Practical Comptrollership Course cannot be expanded. Secondly, it was not possible to eliminate any of the topics from the current course content. Therefore the

microcomputer course must be designed as a tutorial to be accomplished after hours by the students. The environment of the motel room imposes two problems to be overcome. First the student must have the ability to stop at certain points in the course, and be able to return to these points later in the evening. Secondly, the course must be self sufficient in that the student must be able to go through the course without an instructor present.

The various backgrounds of the prospective students and the environment in which the course is to be taken limited the scope and depth of the tutorial. The course had to be basic enough to allow the inexperienced user to gain the knowledge and experience set forth in the course goals. The course, also, must be related to financial management. Therefore it must provide information in both computer and software use within the field of financial management.

C. DEFINITION

Four goals were derived as minimal requirements for the tutorial. These were developed in discussion with the senior PCC instructor and in consideration of the above limitations.

1. Goal One

The first goal of the tutorial is to give the student a basic introduction and familiarization in the operation and use of a modern microcomputer. Since this course would be the initial experience with microcomputers for many of the students, an in-class presentation by the instructor would ease some of their anxieties. This presentation will point out the different components of the microcomputer and the purpose of each. The instructor will

also explain how to plug in the microcomputer, how to turn it on, and some of its basic operations to aid the students in becoming familiar with the machine. The terminology used in describing the equipment and its operation must be precise and consistent in all forms. The instructor and the manual must use the same terminology. Similarly the start up instructions in the manual must agree with the instructions given by the instructor in class.

2. Goal Two

The second goal of the course is for the students to receive a basic introduction into a commercial database package. There are five objectives to meet in accomplishing this goal:

1. Understand the purpose of a database.
2. Understand the makeup and structure of a database file.
3. To be exposed to the creation of a database file, understand the editing features of a database file.
4. Observe the manipulation of files using some of the available commands.

The first objective for the student is to receive a basic explanation of some elementary database theory. This will build a base for the student to expand upon. In accomplishing this objective, it is necessary to describe the similarities between a database and a manual filing system the student has used in the past and how the maintenance of data is now capable of being done electronically.

The second objective is to show the student a breakdown of the different components which makeup a database file. The components which make up the file, such as file name, records, and field names must be explained. The file structure will also be displayed and explained.

This explanation will include how a file is set up, how the fields are defined, the possible field types, and how the length of each field is defined.

The third objective is for the student to define and create a database file. This will be accomplished by the student following step by step directions given in the tutorial manual.

The fourth objective is accomplished through the discussion and presentation of the editing and reviewing modes. The difference between the two modes will be explained. Also included in the discussion will be the various methods to leave the editing and reviewing modes and the purpose of each. The student will then be given a step by step procedure to edit the database file previously created.

The fifth objective is to demonstrate some of the commands which manipulate the database files, such as sorting and joining database files. Examples of each of these commands will be given and the advantages and disadvantages of each will be discussed.

3. Goal Three

The third goal of the tutorial is for the students to receive a basic introduction into commercially available electronic spreadsheet package. There are three objectives to meet within this goal:

1. Develop familiarity with the electronic spreadsheet,
2. Complete a partially completed worksheet, and
3. Use a spreadsheet to analyze "what-if" scenarios.

The first objective of this goal was actually found to be made up of several sub-objectives. For the student to become familiar with an electronic spreadsheet the concept of a manually operated spreadsheet had to be understood. Once this is accomplished the transition to the electronic

spreadsheet is an easy one. The student must understand the different parts of the screen displays of a worksheet, how to build a worksheet, the various options available when building a worksheet, and how to manipulate the data presented. These are accomplished through the use of the Lotus 1-2-3 Getting Started Tutorial. Crawford [Ref. 6: p. 17] suggested that "once the topic has been chosen, you should review courseware which is presently available for material which may meet your requirements to prevent you from having to reinvent the wheel". LOTUS 1-2-3, then, was chosen over building a separate tutorial because it accomplished the desired objectives in a course which would take the most inexperienced user no more than one hour to complete.

The second objective, which is to complete a partially completed worksheet, can be accomplished in one of two ways by the student. The first option will describe the items to be added to the worksheet and the locations at which they are to be added. The student will then make the required additions using the knowledge gained in the previous section of the tutorial. If the student has not gained sufficient knowledge to make the necessary additions a second option will give the student a key by key description of how to make the required additions to the worksheet. The additions to be made to the worksheet include the addition of labels, titles, and formulas, copying these formulas to adjacent columns, and formating columns of data.

The last objective requires the student to apply spreadsheet capabilities to a sample real-life financial management situation by analyzing the effects of budget adjustments in a completed worksheet similar to one they might encounter in the execution of their daily duties. The student will make changes to the budget worksheet and review

the results. This real life situation is taken from an event which the financial managers are familiar. This will allow them to relate their past experiences to the present experience and see how it can work to their advantage.

4. Goal Four

The fourth goal of the course is to remove any initial fear the student may have towards microcomputers. This goal is accomplished through the following objectives. First the tutorial is set up so that after the initial presentation by the instructor in class, the student can take the microcomputer back to the motel room and complete the course without any other outside instruction. Secondly, the student has the opportunity to experiment with system after completing the course without anyone observing them. Being able to complete the course will build up the users confidence. The lack of the outside observations when the students make mistakes will relieve much of their inhibitions to experiment with this new technology [Ref. 6: p. 1].

D. LOCAL STRUCTURE

The first consideration in determining the strategy to be used when developing CAL is to decide upon a pre-structured or adaptive system. Although any of the strategies previously discussed can be implemented with either a pre-structured or adaptive system, some are better suited for one or the other. A pre-structured system was decided upon because of the ease of implementation, the experience level of programmer, and the limited man-hours available for the project. Building an adaptive system would have required additional personnel to complete the project.

The strategy chosen to implement the system was a combination of the inquiry, simulation, and tutorial strategies. The inquiry strategy was used to allow the student to select the particular topic to be covered from a menu when desired. This was chosen for three reasons. First, the students have a wide variety of experiences with microcomputers, therefore not all students will desire or need to complete all phases of the course. Secondly, the students will have the option to take a break between sections and then be able to restart when desired. Finally, the student will be able to repeat a section if needed. The simulation strategy was used during the budget analysis section to illustrate cause and effect situations which develop in the budgeting process. The tutorial strategy was used in the spreadsheet and database lessons, where the student is given a combination of background material and instructions in the tutorial manual and on the screen. Then the student is asked to build and edit a file. This practical exercise is used instead of giving a quiz on the material presented. Feedback to the student from the tutorial results from the student seeing the results as anticipated rather than being told that their inputs were correct.

E. PRESENTATION

The basic presentation will use a tutorial manual and interaction with screen displays The presentation must flow easily for the student who must be able to move between the different sections or media without getting lost or confused. The directions to move between the manual and the screen are designed to stand out from the rest of the information on the screen or in the manual so the student will not be able to miss them.

Using a microcomputer keyboard as an input device is relatively easy for most adults, especially if they have a familiarity with a typewriter keyboard. Special instructions had to be included to explain the special microcomputer keys which are not part of a normal keyboard.

On-screen presentations are displayed as individual pages or screens, instead of accumulating lines of information until they are forced off the top of the screen by the newest entry at the bottom of the screen. Allowing the data to accumulate usually gives a cluttered look to the screen and tends to confuse the reader. By clearing old information from the screen and displaying a new screen, one can control what a user is observing at a given point in time. This also removes any confusion the students may have regarding what they are supposed to be viewing at any particular point in time.

The display format is consistent between the individual screens. When a record or file is displayed followed by text, the file is displayed on the top half of the screen with the text below separated by a line. Each screen contains a complete idea within itself.

All the key-by-key instructions given to the student to use while inputting data are followed by a description of what will happen when the student inputs that data. This was done as a feedback mechanism for the student to reassure them that they were making the correct entries.

Upper and lower case letters were used to make the screens easier to read, since resolution on the screen is not as good as in print [Ref. 6: p. 43]. Upper case statements are used to get the students attention, at points where it was necessary to provide special instructions.

F. SITE IMPLEMENTATION

Site implementation was an on going phase though out most of the project. It consisted of several subordinate phases which progressed to the final product. The first of these phases was hardware and software familiarization. In this phase there was a requirement to become familiar with both the hardware to be used and the software available. The second subordinate phase was the integration of the hardware and software. The final phase was the actual construction of the course based upon the previously designed material.

The selection of hardware for this tutorial was accomplished with the requirements set forth by the senior PCC instructor. These requirements were developed by considering the environment in which the tutorial would be taken by the students.

Due to the requirements of the PCC there is no available time during the normal classroom hours for the students to spend four hours taking the tutorial. Therefore they will be required to take the tutorial after normal class time in their motel rooms. This requires that the microcomputer used for the tutorial be portable.

As previously discussed, the students generally have little experience with microcomputers. This lack of experience dictates that the student be required to do as little as possible in starting up the tutorial. A microcomputer with a hard disk helps solve this problem. The software can be loaded on the hard disk and started up automatically when the system is turned on by the student.

The system which met the above requirements was a COMPAQ microcomputer. The model purchased was a portable system and came equipped with 256k bytes of internal memory, one 360k byte double sided double density floppy disk drive, and a 10 megabyte hard disk drive.

During the development of the tutorial it was discovered that the software being used required more than 256k bytes of memory to process one particular instruction which was required to implement the program as desired. Therefore the steps required to acquire the additional memory were undertaken by the senior PCC instructor.

The hardware to be used in the project had not been received when the familiarization phase was started. Since the COMPAQ microcomputer is an IBM compatible computer, the non-availability of the COMPAQ did not present a problem as there were several IBM microcomputers available for research and development use. Therefore the IBM microcomputers where used in the interim as the hardware familiarization tool. The hardware familiarization consisted of learning the system operation and the disk operating system (DOS). The sources for this information this were the IBM manufacturer's supplied manuals and a DOS tutorial. There were no problems involved in converting to the COMPAQ microcomputers when they arrived.

The software familiarization consisted of a review of the software locally available, completing the training provided with each software package and a literature search to determine the reputation of each. The software packages reviewed were Lotus 1-2-3, RBASE 4000, dBASE II, and dBASE III.

Lotus 1-2-3 is a software package which contained both a spreadsheet and a rudimentary database package. This package comes with built in tutorials, four on the spreadsheet and one on the database. Each of these tutorials take the average user about thirty minutes to complete. The four spreadsheet tutorials start with a simple introduction into the basics of spreadsheets and progress to complex systems.

RBASE 4000, a relational database management package, was the first database software to be evaluated. The package comes with a special tutorial disk which holds predefined database files that the user calls by following the directions in a tutorial manual. The database, although a powerful system, unfortunately requires an experienced user to be able to understand and utilize the complicated commands required to manipulate the system.

Both dBASE II and dBASE III are two database packages produced by Ashton-Tate. It should be noted that dBASE III is an upgraded version of dBASE II which has been the markets leading microcomputer database package for the last two years [Ref. 8: p. 61]. The two systems use basically the same commands to operate, although there were some new commands added and some old commands replaced by new commands when dBASE II was upgraded to dBASE III. The major improvements have been an increase in speed and ability to update multiple-keyed index files. dBASE III also has increased the number of databases and files which can be open at a time. The major advantage of dBASE II or dBASE III over RBASE 4000 is the ability of the command program files to store and automatically run dBASE instructions. This ability allows the user to run commands repeatedly by giving the system a single command. The tutorial which comes with the dBASE systems is similar to the RBASE 4000 package in that it is a lesson which is read from the manual and uses files stored on the disk for the student to call up and manipulate in an on-line fashion.

The integration of the hardware and software entailed the selection of the software packages to be used and installing the software on the COMPAQ microcomputer. The Lotus 1-2-3 spreadsheet was chosen due to the ease of operation and its popularity in both civilian industry and federal government [Ref. 4,8: p. 46,61]. The Lotus 1-2-3

database was not chosen because it did not emulate a standard database system. Rather, it attempted to use the spreadsheet cells to simulate a flat file system with some sorting ability. The database package chosen to be used was dBASE III. This was chosen over RBASE 4000 because of the ability to program command files to run desired commands repeatedly and because of the popularity of the dBASE II/III systems among users.

The tutorial was constructed based upon the previously discussed definition layer, local structure, and presentation. The dBASE III command language was chosen as the programming language due to its versatility, clarity and maintainability. It uses all the commands and functions directly from the dBASE III language which make the programs easily understood and changed by anyone with an average knowledge of these commands.

The program is centered around the program Main.prg. This program prints the menu to the screen which gives the student their different options and interprets their selections. The program then, based upon the students selection, calls one or more programs to present the lesson requested. The sub-programs available to be called are Spread.prg, Dbase.prg, and Budget.prg as shown in Figure 4.1.

Spread.prg is called when the student selects the spreadsheet option. This program causes two Lotus 1-2-3 programs to be called, the tutorial and then the spreadsheet. The student works the Getting Started tutorial, and then completes the construction of an actual spreadsheet. Upon completion, this program calls the Main.prg to display the main menu.

Dbase.prg is the first of four programs which are called when the student selects the database lesson. These programs work through the various aspects of the lesson,

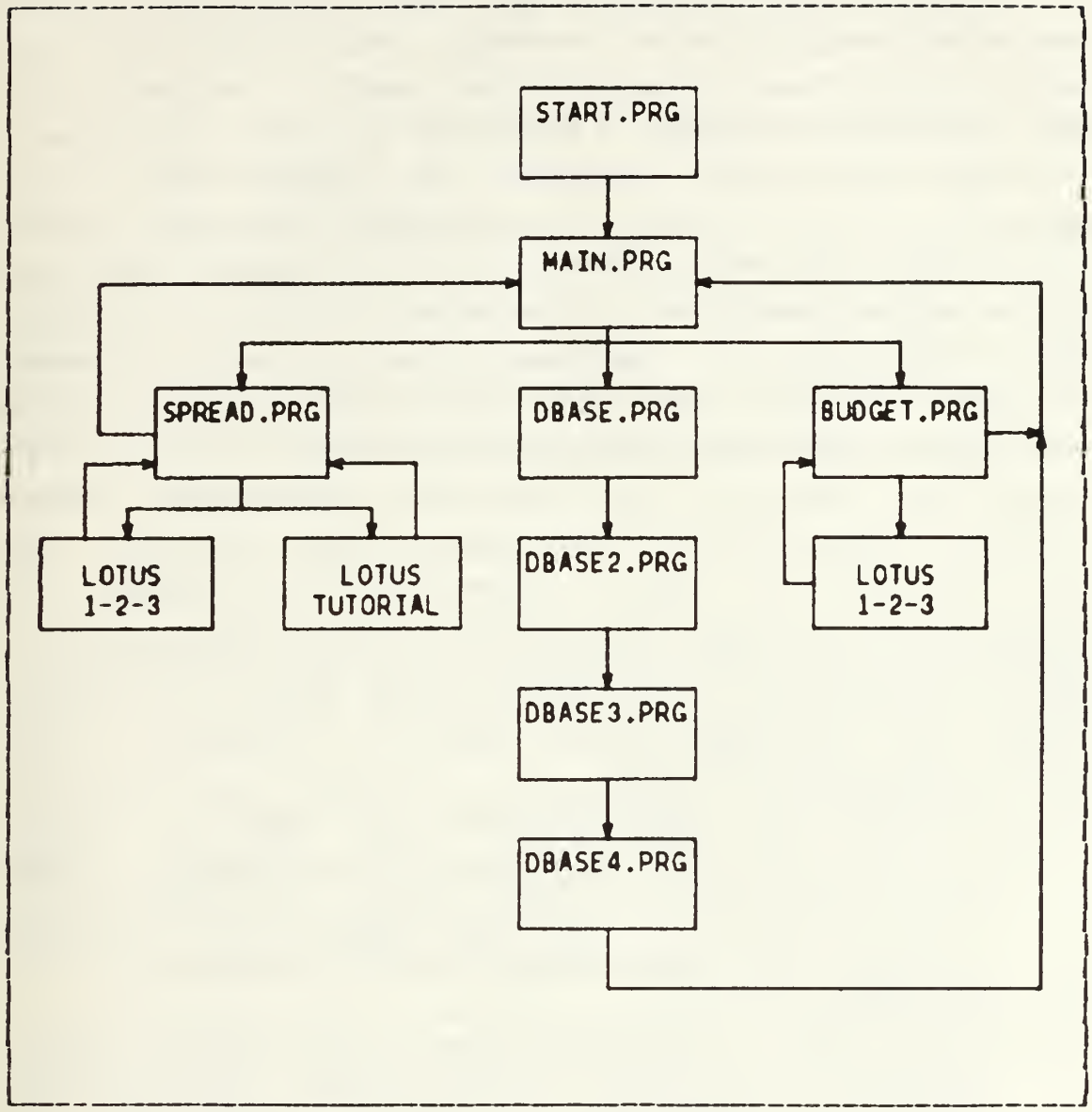

Figure 4.1 Program Modules

calling the different database files and command programs. Upon completion of these programs, Main.prg is called to display the main selection menu.

Budjet.prg is called when the student selects the budjet analysis option. This program calls the Lotus 1-2-3 spreadsheet and the student works with changing values in the spreadsheet cells.

When the student selects the "Quit" option the system erases the temporary files created by the student, exits the program and automatically parks the heads on the hard disk. This is to prevent possible damage to the hard disk system.

Each part of the program was designed and tested separately prior to being implemented into the main program. The tutorial was then tested with three students from each of the February and March PCC courses. This test required that the students take the tutorial with a monitor present to answer questions which they might have. The students were given a short introduction by the monitor which quickly covered the topics the PCC instructor would cover in the in-class presentation. The students were not given the details which the in-class presentation would cover to simulate the deterioration the information the students would experience by the time they were back in the motel room. The following changes resulted from major points brought out by the testing:

1. Directions on the screen for the students to return to the manual must be on a page by themselves. This change was made because these directions where usually missed when they were combined with any other display.
2. There must be a definite break in the tutorial manual between the text and directions for inputting data into the microcomputer. This resulted from the students reading past the directions, thinking they were part of the text.
3. When data is presented to the student for example purposes only, this must be so stated and made to stand out from the rest of the text. If this was not done most of the students would attempt to input the data and would encounter problems.

4. Terminology was not consistent throughout the tutorial manual. The most frequent inconsistency was the use of "Return" and "Enter" as synonymous terms. This caused much confusion among the students, and subsequently, the terminology has been changed to a consistent "Enter".

Upon completion of the tutorial the six students were all looking forward to using the microcomputers at their commands to continue their training. All felt that it was at a level which did not make them feel intimidated or lost. None of the students appeared to become flustered while taking the tutorial.

G. SUMMARY

The experience of the potential users of the tutorial limited the depth of the goals which could be implemented into the tutorial. The four goals, which are an overview of microcomputer operations, databases, spreadsheets, and removal of any fears the students may have towards microcomputers, were covered at a level which reflected the students lack of experience.

A combination of the instructional strategies was used to develop the local structure to maximize the benefits the students would receive from the tutorial. These strategies were then combined together in a pre-structured design.

The presentation of the tutorial using both the tutorial manual and on screen presentations gives the student the ability to read about the topics before having to work with the microcomputer. The tutorial manual also gives the student the capability to go back and review material when presented a situation on the screen of the microcomputer which may not be understood. Step-by-step instructions provided in the tutorial manual allow the student to use the

detailed instructions if they desire, but also allow the experienced student to make the entries without using the instructions.

Implementation of the previous steps included the software analysis and the integration of the software and hardware. The development of the software to implement the project was broken down into modules which were connected together after they were individually designed and testing was completed.

Based on the testing accomplished, the tutorial appears to have achieved the goals set for the project in Chapter 3, although, achievement of these goals must be continually monitored as future PCC students take the tutorial to ensure that they are still valid. As the knowledge of the general population in the field of computers increases in the future the goals will require modification to reach out to a higher level of training.

V. CONCLUSIONS

A. INTRODUCTION

The tutorial this project has developed will fill a void which currently exists in the Navys training of its financial managers. In prototype testing it has shown itself to be successful in meeting its proposed goals with the typical users of the course.

B. DISCUSSION

The overall goal of this project was to develop a tutorial that could be used to improve the Navy financial manager's knowledge level and experience in using microcomputers. The tutorial was designed with the prospective students in mind, giving consideration to the fact that most of them had no experience with microcomputers or modern software packages.

The layered methodology used to develop the tutorial presented an orderly and smooth flowing project. The separation of the project into layers ensured that each layer was developed without being influenced by the lower layers. The methodology used was independent of the tutorial material, software, and hardware used. This makes it ideal to be used on any CAL development project.

The software used to write the programs, dBASE III, was adequate for the type of tutorial which was developed. It would be of limited use in a highly interactive or adaptive system. It provided an excellent sampling of database operations and functions for the tutorial. LOTUS 1-2-3 is also an excellent spreadsheet to teach from because of its user friendliness. Other tutorials which could be

developed, using the same methodology and software packages, for the PCC students should present an advanced version of this tutorial for the students experienced in databases and spreadsheets.

It is recommended that this tutorial, which is included as Appendix A, be given to all future students of the PCC. This would help improve the microcomputer, spreadsheet, and database knowledge level of the Navy's financial managers. Although not all financial managers attend the PCC, each class is made up of financial managers from a wide cross-section of different commands. Therefore, the students who become familiar with the material in the tutorial will carry back to their various commands a knowledge which will help the command when they are required to implement a microcomputer based software package.

APPENDIX A
TUTORIAL GUIDE

A. GETTING STARTED

If you are using a microcomputer for the first time, don't be afraid of this tutorial or the microcomputer. You won't break the micorcomputer or harm the tutorial program. This guide was written for the person who has never touched a computer before taking this course. If you press the wrong key the computer will not tell anyone else. So forget any fears you have of computers and sit back and enjoy the learning experience you are about to have.

1. Specifications

For those of you that are familiar with microcomputers, the COMPAQ portable microcomputer you going to use has a 512 K-byte random access memory (RAM), a single 5 1/4 inch, double-sided, double-density, 360 K-byte floppy disk drive, and a 10 Megabyte hard-disk drive. It runs on normal house current, 115-120 volts, 60 cycles. The power cord has a three prong plug, which must be connected to a three pronged wall socket via the surge suppressor to ensure a proper connection for electrical safety. The surge suppressor is designed to prevent fluctuations in the electrical power causing electrical damage to the microcomputer.

2. Setting Up

Set the computer down, with the handle up, on the desk top you will be working upon. On one of the large faces you will find two feet. Open them up by flipping them

out towards the side. These feet will elevate the front of the microcomputer to allow easy viewing of the video screen.

Lay the microcomputer down on its feet with the handle away from you. Now you have what was the bottom of the microcomputer facing you. Note a second pair of feet in the lower right and left-hand corners. Open them the same way you opened the support feet for the main body of the microcomputer. Also note the two nylon latches on the right and left sides above the two feet. These secure the base, which is also the keyboard, to the microcomputer. Hold on to the base so it does not fall, and open the nylon latches by sliding them towards the center. Don't worry, the base is not heavy. Lower the keyboard to the desk top in front of the microcomputer.

On the left hand side of the microcomputer you will find a sliding panel. Place two fingers into the finger hold at the back of the panel and your thumb at the front of the panel. Now press in with your thumb and forward with your fingers, to slide the door forward into the casing.

In the power compartment that is now exposed you will find the power cord, its connection point to the microcomputer, the on-off switch, and the ventilation fan. Remove the power cord from the compartment.

Ensure the on-off switch is in the off position; the bottom of the switch should be depressed. Plug the cord into the microcomputer just below the on-off switch, then plug it into the surge suppressor. The surge suppressor should then be plugged into a three pronged wall outlet.

Check the floppy disk drive to ensure that there are no disks in the drive. Do this by flipping the lever on the front of the floppy disk drive to the vertical position. If a disk is currently in the drive it will pop out about one inch. If there is a disk in the drive, remove it and place it aside to return it to the instructor. Leave the lever in the vertical position.

3. Power-up

Now turn the power on by pushing the on-off switch (located in the power compartment on the left hand side of the microcomputer) in at the top. With power applied to the system the ventilation fan will start, the hard disk drive will spin up to speed, and the system will do some internal memory checks. You will see a green light on the hard disk drive flicker a couple of times and then come on steady. Also you will notice the red lights on both the floppy and hard disk drives go on and off a couple of times. On the screen you will see the message "1 HARD DRIVE". Finally after about 25 seconds everything will settle down and on the screen you will observe:
Current date is Tues 1-01-80 Enter new date:

4. Initializing the Machine

At this time take the floppy disk, provided to you by the instructor, remove it from its protective sleeve, and then insert the disk into the floppy disk drive. This is accomplished by holding the disk in the vertical direction by the black cover, with the label on the left side closest to you, as shown in Figure A.1. The long opening in the black cover should be in a horizontal position, on the side away from you. Never touch the actual disk where it is visible through the cover. Slip the disk into the drive until it stops with a click, then flip the lever to the right over the disk, to the horizontal position.

Now back to the screen and the date. Enter the date in the following format:

MM-DD-YY as in 12-25-84

Then push the enter key, which is located between the typing keys and the numerical keypad, just to the right of the "]"

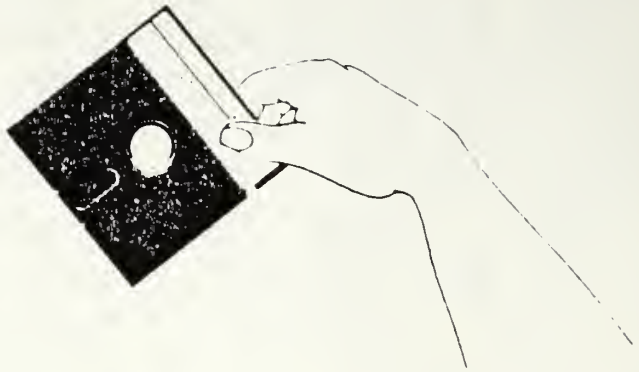

Figure A.1 Inserting floppy disk
Reproduced from Compaq Manual.

keys. You will then observe the on screen tutorial introduction, as shown in Figure A.2.

```
                    TUTORIAL SELECTION

            1 => SPREADSHEET TUTORIAL
            2 => DATABASE TUTORIAL
            3 => BUDGET ANALYSIS
            Q => QUIT


     CHOOSE 1,2,3 OR Q
```

Figure A.2 Main Menu

5. Getting Out of the Tutorial

If you wish to leave the tutorial at any time, you must get back to the main menu and select the quit option.

When you select the quit option the system will change any files you have modified as part of the tutorial back to the original form and erase any files you have generated. It will also run a program which will park the heads of the hard disk drive in a safe position. This is very important as the machine could be damaged when being transported if the heads were not parked.

Below is a list of how to leave the tutorial from the different modes.

At the MAIN MENU

Select "Q" to quit.

In the SPREADSHEET TUTORIAL (LOTUS 1-2-3)

Press: / (to bring up the menu into the control panel)
Press: Q (to select the quit option)
Press: Y (to confirm that you want to quit 1-2-3)

You will now have the main menu on the screen. Select "Q" to leave leave the tutorial.

In DATABASE TUTORIAL

Turn the microcomputer off at the power switch and leave the power off for one minute. Then turn the microcomputer on again. When the system requests you enter the new date, just press the enter key. Now when the Main Menu returns to the screen, select "Q" to quit, and turn the power off.

In BUDGET ANALYSIS

While in the budget analysis section of the tutorial you will make an exit the same as described above in the spreadsheet section.

If for some reason the tutorial starts to malfunction and will not allow you to input from the keyboard, it is necessary to turn the power off and restart the tutorial. It is essential that you restart the tutorial

as you have just done and select Q from the main menu shown on the screen, before you move the machine to ensure the heads are properly parked.

B. MAKING A SELECTION

Now its time to begin working with the microcomputer. You have four selections to choose from.

1. Press 1 on the top row of keys for the spreadsheet tutorial. The readings for this lesson start in Paragraph C.
2. Press 2 on the top row of keys for the database tutorial. The readings for this lesson start in Paragraph E.
3. Press 3 on the top row of keys for the budget analysis tutorial. the readings for this lesson start in Paragraph D.
4. Press Q on the letter keyboard to quit the tutorial.

C. ELECTRONIC SPREADSHEET USING LOTUS 1-2-3

1. Introduction

The purpose of an electronic spreadsheet is the same as a manual spreadsheet. It is an organized method of recording financial data in a matrix of columns and rows which are easily summed into totals for the categories of the columns or rows. The major difference is that the electronic spreadsheet will calculate sums of columns or rows, or subtotals or totals of any other combinations of the matrix positions that have been programmed into the spreadsheet, automatically each time a single item on the spreadsheet is initially entered or changed. Another difference, and major advantage of the electronic spreadsheet, is that it does all the arithmetic without making a mistake. But beware, a new electronic spreadsheet or one that has been changed should not be used before it has been properly tested. The spreadsheet will make the calculations it has been told to make, which are not always what you really want it to calculate. Errors in programming the formulas into a new spreadsheet can occur very easily.

The electronic spreadsheet, as is the familiar manual spreadsheet, is nothing more than horizontal rows and vertical columns of data. Each position in this matrix is identified by the column and row at which it is located. These individual positions, or addresses are called cells. The worksheet of Lotus 1-2-3 has 256 columns across labeled by the letters A through IV, and 2048 rows down labeled by the numbers 1 through 2048. A specific cell, then, can be identified by the intersection of the column and row. An example would be the cell in the upper left-hand corner (A1) or the cell in the lower right-hand corner (IV2048). The widths of the columns can be varied from 1 to 72 characters, with the normal width being 9 characters.

2. Display Screen

When working with an electronic spreadsheet, such as shown in the Figure A.3, you view on the screen a small window of the 256x2048 matrix. This window views 20 rows down and as many columns across as will fit on the screen depending upon their assigned width. The window displayed on the screen can be moved to different parts of the worksheet by simply moving the cell pointer.

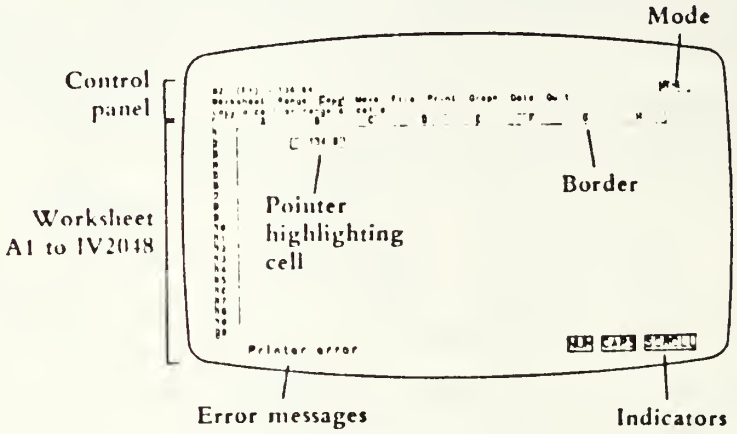

Figure A.3 Spreadsheet Screen

The normally displayed screen is made up of two distinct areas; the control panel and the worksheet, separated by the border area, as shown in Figure A.3.

The control panel, Figure A.4, indicates the status of what the system is doing, what you can do from where you are, and summaries of what you currently have selected. Line one tells you the status of the current cell. Specifically it tells you the address of the current cell, display format, protection status, contents of the current cell, and a mode indicator at the far right of the screen. The current cell address is indicated by letters to indicate the column and numbers to indicate the row. The display format indicates the selected appearance of numbers in the cell. Some of the formats possible are; date, time,

percent, currency($), scientific notation, fixed decimal(0-15 places), and text. The protection status tells you whether the content of the cell is protected, which means it may not be changed unless the protection is removed. The content of the current cell displays what is in the cell. This will display the numeric value or character label if that is what has been placed in the cell. If a formula has been programmed into the current cell, the cell content in the command line displays the actual formula, while the result of the formula will be displayed in the cell location of the worksheet.

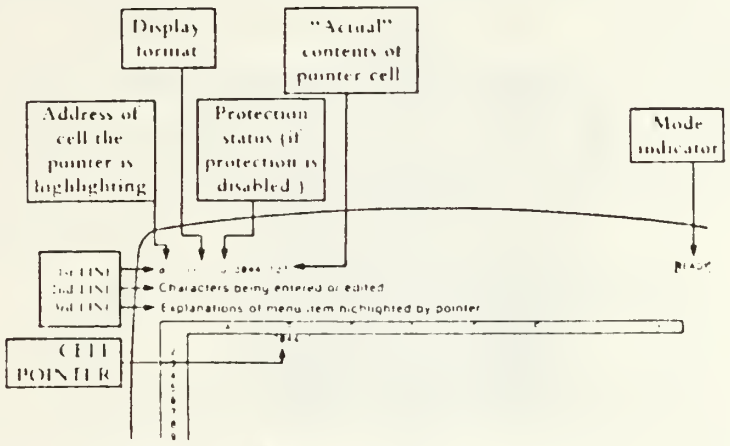
Display format
"Actual" contents of pointer cell
Address of cell the pointer is highlighting
Protection status (if protection is disabled)
Mode indicator
Characters being entered or edited
Explanations of menu item highlighted by pointer
CELL POINTER

Figure A.4 Control Panel

The second line of the control panel changes as you select different options. If you are placing an entry into a cell it will show you what you are typing. Once you push the enter key, this entry will move to the cell contents of line one and into the cell location in the worksheet. If you are making changes to the spreadsheet this line will indicate the options and provide various prompts.

The third line will display small summaries of the various commands you have selected in the second line of the control panel.

When you work through the on-screen tutorial watch the changes that occur in the control panel as you select

different items and push different keys. It is easy to get locked in on the bottom half of the screen where the instructions will appear and miss all the action up here.

The worksheet, Figure A.5, is the working section of the spreadsheet. The current cell of the worksheet is indicated by a reverse-video rectangle called a cell pointer. That is the background of the rectangle is green and the letters or numbers are black. As you select a different cell the cell pointer moves to the newly selected cell.

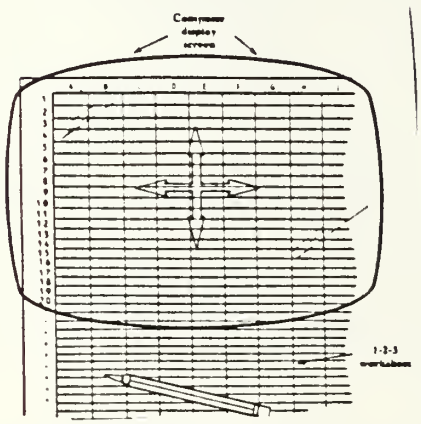

Figure A.5 Worksheet

The individual cells can display labels that you have programmed in (such as 1st QTR, TOTAL, NET), numeric values which you have programmed into the cell (such as 15, 37364), or the results of a formula you have programmed into the cell. The actual formula will be displayed in line one of the control panel.

The border is located between the control panel and the worksheet and down the left side of the screen. The border indicates the range of the current window of the worksheet being displayed, as shown in Figures 1 and 2. The border section between the control panel and the worksheet

indicates the visible vertical columns with the letters A through IV. The border section along the left side of the screen indicates the 20 visible horizontal rows with the numbers 1 through 2048.

3. Tutorial

The spreadsheet tutorial has three separate parts. The first is an introductory tutorial that comes with the Lotus 1-2-3 package. This will take you through the basis of spreadsheet operations. The second part will give you the opportunity to complete a partially built spreadsheet, adding both labels and numbers to it. The third section, which is also listed under Budget Analysis on the main menu, will allow you to enter changes, or marks, to a submitted budget. With this introduction into spreadsheets we will go to the computer for the tutorial.

****
* GO TO SCREEN NOW *
****

4. Completing a Spreadsheet

Every spreadsheet you use will require some initial preparation before you can use it. As you saw in the Lotus tutorial you must put in labels, formulas and formats. Then you must enter the initial set of data which the spreadsheet will hold, formulate, and summarize.

In the Lotus tutorial you were forced to make the correct entry, which probably reduced your fears of making a mistake. In the following sections you will be making some of the same entries as before, but you will be working on an actual spreadsheet, which will allow you to make any entry you wish. Do not let this worry you, as you will be given two options in making these entries. You will be shown what

the required entry is and where it is required. If you wish, you can then make the entry on your own, and then check it with the detailed instructions that follow. If you would rather not attempt to try the entries on your own, there will be detailed instructions for you to follow.

While proceeding through this section you will use the computer and this manual together to make the entries to the spreadsheet.

Now,lets go back to the computer, Lotus 1-2-3 will be called up to the screen for you when you press any key. You will have to press a key to get the spreadsheet up to the screen.

Once you have the spreadsheet up on the screen you will have to call up the partially formatted worksheet from memory. This is done in the following manner:

Press: / (to bring up the menu into the control panel)
Press: F (to select file operations)
Press: R (to retrieve a file)
Move the reverse video area of the control panel cursor to the right to Budgetx
Press: Enter (to retrieve the worksheet called Budgetx)

When the worksheet comes to the screen the cell pointer should be in cell A1. If it isn't, press the "Home" key which is located on the "7" key on the numeric keypad.

Now you are ready to complete the worksheet. Below are two columns which list the cell number and what is to be input to that cell. Move the cell pointer to the proper cell and type in what is between the quotation marks. Remember, there are detailed instructions below for you if you would like some help.

CELL NUMBER	INPUT
B3	The label "MISSION".
B4	The label "OPERATIONS"
B18	The formula to add up cell positions

	B6 thru B16
B20	The formula to subtract cell position B19 from B18.
B18	Copy the formula from cell position B18 to cells C18-N18.
B20	Copy the formula from cell position B20 to cells C20-N20.
B18-Q20	Format these cells to currency format without any decimals.

If you wish you can complete the worksheet from the above information. Otherwise follow the instructions given below.

Press: "Caps Lock" key.
Move the cell pointer to cell B3.
Type: MISSION (label for cell B3)
Press: Enter
Move the cell pointer to cell B4.
Type: OPERATIONS (label for cell B4)
Move cell pointer to cell B6.
Press: / (to select the menu in the control panel)
Press: W (to select the worksheet options)
Press: T (to select the title options)
Press: B (to lockout both the titles on the top and left side.)

You will now enter the formula to add up cells B6 through B16 and place the sum in cell B18.

Move the cell pointer to cell B18.
Press: "+"
Move the cell pointer to cell B6.
Press: "+" (the cell pointer will return to cell B18)
Move the cell pointer to cell B7.
Press: "+"
Move the cell pointer to cell B8.

Press: "+"

Move the cell pointer to cell B9.

Press: "+"

Move the cell pointer to cell B10.

Press: "+"

Move the cell pointer to cell B11.

Press: "+"

Move the cell pointer to cell B12.

Press: "+"

Move the cell pointer to cell B13.

Press: "+"

Move the cell pointer to cell B14.

Press: "+"

Move the cell pointer to cell B15.

Press: "+"

****

NOTE

****

After moving to the next cell (B16) do not push the "+" key.

Move the cell pointer to cell B16.

Press: ENTER

Be sure that you pushed the "Enter" key the last time instead of the "+" key. The formula string should end "....+B14+B15+B16" If you pushed the "+" key after moving the cell pointer to cell B16, you will have to press the "Esc" key and start the entry of the formula over again.

You will now enter the formula to subtract the entry in cell B19 from the sum in cell B18 and place the result in cell B20

Move the cell pointe o cell B20.

Press: "+"

Move the cell pointer to cell B18.

Press: "-" (make sure that you used the minus key located at the extreme right side of the keyboard)

Move the cell pointer to cell B19.

Press: ENTER (as before, if you pressed a "+" or "-" key here you will have to press "Esc" and start this formula over again)

Move the cell pointer to cell B18.

You will now copy the formula in cell B18 into cells C18 through O18.

Press: / (to bring up the menu in the control panel)

Press: C (to initiate the copy routine)

Press: Enter (to utilize the "from range" of B18..B18)

Type: C18..O18 Ensure you put the two periods between the 8 the letter O. Note that LOTUS 1-2-3 will not accept your formula if type the number zero in place of the letter O. (to copy the formula in cell B18 into cells C18 thru O18

Press: ENTER (formula now copied)

Move the cell pointer to cell B20.

You will now copy the formula in cell B20 to cells C20 through N20.

Press: / (to bring up the menu in the control panel)

Press: C (to initiate the copy routine)

Press: ENTER (to utilize the "from range" of B20..B20)

Type: C20..N20 (to copy the formula in cell B20 into cells C20 thru N20)

Press: ENTER (formula now copied)

You will now enter the formula to subtract the value in cell P19 from the value in cell O18 and store the result in cell Q20.

Move the cell pointer to cell Q20.
Press: "+"
Move the cell pointer to cell O18.
Press: "-" (make sure that you used the minus key located at the extreme right side of the keyboard)
Move the cell pointer to cell P19.
Press: ENTER

You will now format the summary rows on the bottom of the worksheet using the currency format without any decimals.

Move the cell pointer to cell B18.
Press: / (to bring up the menu into the control panel)
Press: R (to select the range function)
Press: F (to select the format function)
Press: C (to select the currency format)
Press: 0 (to select zero decimal places)
Press: ENTER (to input the zero decimal places)
Type: B18..Q20 (to define the range of the currency format)
Press: ENTER (to input the typed in range)

You have now completed the worksheet. Check your results with Figure A.6 at the end of the Budget Analysis section. To test your spreadsheet to see if all the formulas you have entered are correct, input a few numbers into various cells inside the rectangle bounded by cells B6 thru N6 and B16 thru N16. Then check the sums in the three bottom rows and the three right columns.

Thats it on spreadsheets for now. To get back to the Main Menu follow the instructions listed below. You may

want to try the "what-if" scenarios in the Budget Analysis option on the Main Menu next.

Press: / (to bring up the menu to the control panel)

Press: Q (to select the quit option)

Press: Y (to confirm that you want to quit 1-2-3)

You will now have the Main Menu on the screen as shown in Figure A.2.

D. BUDGET ANALYSIS

In this section of the tutorial you will apply the techniques you learned in the spreadsheet tutorial to a sample real-life financial management situation by analyzing the effects of budget adjustments in a completed worksheet similar to the one you might encounter in the execution of your normal duties. You will make changes to the budget worksheet and review the results. While proceeding through this section you will use the computer and this manual together to make the adjustments to the spreadsheet.

Now lets have some fun! Here is the scenario.

You have just come into work and saw the "big black eye" of your desktop computer staring at you. Yes, it wasn't too many weeks ago when you felt like a slave to that machine while preparing your budget submission. Those were terrible times. You can remember being caught between the "control numbers" given to you by your major claimant and the desires of your commanding officer to improve performance in two of the functional areas at your command. The cries for more resources by those cost centers still ring in your ears. If it wasn't for your electronic spreadsheets and trusty ole Lotus 1-2-3, the budget would never have been finished in time. It sure would be nice to have a nickel for each time a spreadsheet was revised during the last two weeks before the budget submission.

Well, enough with the reminisicing...lets look at the morning message traffic. Next page please.

ROUTINE

PT0005 023 003245 PAGE 01

RTTUZYUW RUCKSGG1738 0230029-UUUU--RUWJAGA

ZNR 211948Z APR 85

FM MY MAJOR CLAIMANT WASHINGTON DC

TO RUWJAGA/MY COMMAND MYTOWN ST

BT

UNCLAS //N07110//

SUBJ: REVISION OF COMMAND CONTROL NUMBERS

A. MTG BTW CNO (92) AND MY MAJOR CLAIMANT OF 19 APR 85

B. BUDGET SUBMISSION OF MY COMMAND OF 22 MARCH 85

1. AS A RESULT OF REF A, FUNDS AVAILABLE TO THIS CLAIMANCY HAS BEEN REDUCED. SIGNIFICANT REDUCTIONS HAVE BEEN EXPERIENCED IN ELEMENT OF EXPENSE "U" (CIVILIAN PERSONNEL) WITH LESS SIGNIFICANT REDUCTIONS OCCURING IN ELEMENTS OF EXPENSE "E" (TRAVEL OF PERSONNEL), "Q" (OTHER PURCHASED SERVICES), "W" (EQUIPMENT) AND "V" (OTHER POL). IT IS REGRETTED THAT BUDGET MARKS RECEIVED BY THIS COMMAND MUST BE ABSORBED BY SUBORDINATE COMMANDS SINCE INADEQUATE UNDISTRIBUTED RESOURCES REMAIN AT CLAIMANCY LEVEL TO ABSORB THE MARKS.

2. ACCORDINGLY MY COMMAND IS ASSESSED THE FOLLOWING REDUCTIONS:

ELEMENT OF EXPENSE	REDUCTION	NEW CONTROL
CIVILIAN PERSONNEL (U)	$500,000	$28,055K
TRAVEL OF PERSONNEL (E)	10,000	445K
OTHER PURCHASED SERVICES (Q)	155,000	6,029K
EQUIPMENT (W)	18,000	658K
OTHER POL (V)	3,000	60K

3. IT IS REQUESTED THAT MYCOMMAND SPREAD THE REDUCTIONS IN THE FUNCTIONAL CATEGORIES IN A MANNER THAT WILL MINIMIZE THE OPERATIONAL IMPACT UPON MYCOMMAND. FURTHER REQUEST SUBMIT REVISED NAVCOMPT FORM 2179-1, ACTIVITY BUDGET/APPORTIONMENT SUBMISSION TO THIS COMMAND NLT COB 27 APR 85.

BT

Oh, No! Here we go again. It doesn't seem possible to absorb a budget cut of this magnitude. How can this be spread out among the departments? What is the impact on the budgets of the various cost centers? Can it be spread through an across-the-board cut? That would be the easiest and quickest way to distribute the reduction, but it fails to recognize the CO's desires to beef-up the mission operations (M8MZ) and the maintenance of real property (F4FA) cost centers. What is the new total budget amount? Guess I had better get some answers before the morning message meeting. Time to fire up the "big black eye"!

Now, lets go back to the computer, LOTUS 1-2-3 will be called up for you when you press any key. You will also have to press a key to get the blank spreadsheet up to the screen.

Once you have the blank spreadsheet up on the screen you will have to call up the completed worksheet from memory. This is done in the following manner:

PRESS: / (to bring up the menu into the control panel)
PRESS: F (to select file operations)
PRESS: R (to retrieve a file)
Move the reverse video area of the control panel cursor to the right to Budget4.
PRESS: ENTER (to retrieve the worksheet called Budget4)

When the worksheet comes to the screen the cell pointer should be in cell B6. If it isn't, press the "Home" key which is located on the "7" key on the numeric keypad.

Instructions

The categories along the top of the spreadsheet are the SAG/FC/SFC that are utilized by the command. The rows each represent one element of expense. The task is to spread the budget marks within any given element of expense across the SAG/FC/SFC's such that the total for any given element of

expense comes within the control totals given in the message above. For example, one could reduce the value in cell C7 from 1197 to 697 and absorb the total civilian personnel decrease within the administration (F3FF). But it is easy to see from the bottom line that this would have an adverse effect on this functional area. There are innumerable possibilities that exist to affect this reduction...some alternatives are better than others. The thing that you must watch for is the cumulative effects of several adjustments (i.e. Administration may be able to absorb a part of the civpay mark or all of the travel of personnel mark but the cumulative effects of taking both may adversely affect the total budget in that functional area). Play with the numbers by entering your adjustments in the rectangle bounded by cells B6 to N6 and B16 to N16. Examine the effect of your adjustments and ensure that your total budget in the bottom right hand corner of the spreadsheet (cell Q20) reflects the cumulative total of the marks received in the message above and that each total in the right-hand column (total direct expenses) is within the control totals specified in the message. Good luck! When you are finished you will need to get to the main menu by doing the following:

PRESS: / (to bring up the menu into the control panel)
PRESS: Q (to select the quit option)
PRESS: Y (to confirm that you want to quit 1-2-3)

You will now have the main menu on the screen as shown in Figure A.2.

ELEMENTS OF EXPENSE	M8MZ MISSION OPERATIONS	F3FF ADMINIS-TRATION	F3FG SUPPLY OPERATIONS	F3FN MAINTENANCE OF MATERIAL	F3V2 PROPERTY DISPOSAL	F3FR MEDICAL OPERATIONS	F4FA OVERSEAS DEPENDENT EDUCATION	F3FC BASE SERVICES	F3FD MAINTENANCE OF REAL PROPERTY
CIVILIAN PERSONNEL	15662	1197	548	15	3[illegible]	586	1562	97	641
TRAVEL OF PERSONNEL	489	39	2						3
TRANSPORTATION OF THINGS	18	8							54
UTILITIES AND RENTS	293	228	2					1366	18
COMMUNICATIONS	26			547					
PURCHASED EQUIP MAINT	618	37	33		1				3
PRINTING & REPRODUCTION	38	2	2						2
OTHER PURCHASED SERVICES	2788	145	4				1261	152	732
OTHER SUPPLIES	349	188	12		54	44	267	11	58
EQUIPMENT	611	23	13				6	1	15
OTHER POL	2					68		1	
TOTAL OPERATING EXPENSES	$28,688	$1,771	$688	$562	$379	$618	$3,896	$1,628	$1,518
LESS ANTICIPATED REIMBURS									
TOTAL DIRECT EXPENSES	$28,688	$1,771	$688	$562	$379	$618	$3,896	$1,628	$1,518

ELEMENTS OF EXPENSE	F4FB OPERATION OF UTILITIES	F3FK OTHER ENGINEERING SUPPORT	F3FJ MINOR CONSTRUC-TION	F3FL PERSONNEL SUPPORT	TOTAL OPERATING EXPENSES	LESS: ANTICIPATED REIMBURSE-MENTS	TOTAL DIRECT EXPENSES
CIVILIAN PERSONNEL	85	15		21	28666		28666
TRAVEL OF PERSONNEL		2			455		455
TRANSPORTATION OF THINGS					72		72
UTILITIES AND RENTS					1899		1899
COMMUNICATIONS					573		573
PURCHASED EQUIP MAINT					692		692
PRINTING & REPRODUCTION					36		36
OTHER PURCHASED SERVICES	748	284	8	238	6184		6184
OTHER SUPPLIES	85	19	23	43	1857		1857
EQUIPMENT		7			676		676
OTHER POL					63		63
TOTAL OPERATING EXPENSES	$918	$247	$31	$294	$32,262		
LESS ANTICIPATED REIMBURS						$0	
TOTAL DIRECT EXPENSES	$918	$247	$31	$294			$32,262

Figure A.6 Completed Spreadsheet

E. DATABASE TUTORIAL INTRODUCTION

Welcome to the database tutorial. This lesson will give you a peek into the world of databases. It will attempt to cover what a database is, what it can do for you and some of the mechanics of using one.

The database tutorial will use both on-screen displays and this guide to supplement what is shown on the screen. As stated above no prior knowledge or experience is required to use this tutorial.

You will be instructed to go back to the computer at various points in this guide to observe a screen display or input data to the computer. When this occurs, leave the guide open to the page you were at. You will want to return to that point in this guide when instructed to by the screen displays.

1. Purpose of a Database

The basic purpose of a database is to maintain data in an organized and logical manner in a single location. The concept is the same as keeping small records such as name, address, job, department, etc, in a card file or maintaining the extensive personnel files of a large agency or corporation in many filing cabinets. The database, which can be viewed as an electronic filing cabinet, will allow an almost instant examination of various links or relationships between disparate data items and records. The same links could be accumulated in a manually operated paper filing system, but would require an extensive manual search and comparison for the requested data at a rate much slower than the millions of comparisons per second the computer is capable of accomplishing. The manual system would have a certain percentage of errors. The electronic database, once it has been properly tested, will have no errors.

2. Tutorial

a. Description

A database is made up of files, records, fields and characters, with each being a subset of the previous item. This is actually no different than the way information is stored in a manual system. A company may have files of personnel records. These files are made up of individual personnel records which contain specific items or fields of information about the person whom the record is describing such as name, address, age, etc. Each of the data fields are made up of characters.

Different types of database systems have different names for the parts of a database. A file may be called a table. An attribute may refer to a field. A tuple may be a record. These different names need not worry you. When you start to work with a particular database system the names given each part of the database will be explained in the database manual.

The database software system this tutorial uses, dBASE, is described as a relational database. That is, the data it represents is presented as tables. The entire table represents a complete file. Each horizontal row of the table represents a specific record. The table is broken up into columns of various widths, with each column representing a specific field.

Now we will look at a database file I have made up for you.

```
**********************************
*      GO TO THE SCREEN NOW      *
**********************************
```

b. File Structure

Each field of a database record is defined by the designer of the database. The definition limits the data which is stored in each of the fields of the database file being defined. The defining of a field includes the name of the field, the type of the field, its width, and if required the number of decimal places. In the file STUDENT, ID_STU is the first field name. ID_STU represents the identification number of a student. It is a numeric field, four numbers in width. The rest of the fields are all character fields of various widths, except for BIRTH_DATE field, which is a date field. A character field may contain characters or numbers, but you can not perform mathematical manipulations upon the numbers. However, if you had a field listing a persons hourly pay rate which was defined as a numeric field, you could have the database sum all the employees hourly pay rates and determine what it costs to run the company for an hour. If the pay rate was defined as a character field the database program could not do this for you.

Lets go take a look at how the file STUDENT is structured. When looking at the file structure on the screen you may want to refer to the explanation in the above paragraph.

```
**********************************
*     GO TO THE SCREEN NOW       *
**********************************
```

c. Creating A Database File

Creating a database file using dBASE III is relatively simple as most of the work is done for you by the program. You simply type in the data as it is requested on the screen.

The most difficult part in developing a database is determining what should go into each record in a file. It would be very easy for the designer of a database to make a file which contained a field for everything you wanted to know about someone or something. An example of this would be to build a file of records on students and include in each record all of the following:

Student ID number
Last name
First name
Middle initial
Date of birth
Major
Class
Grade average
Course #1
Course #2
Course #3
Course #4
Activity #1
Activity #2
Grade average
Financial aid status
Local address
Home address

There are many more items you could add to the list. This shows the first problem with putting everything into one file - if you want to add more data about a student you must add more fields to the file structure, and this must be done for every record within this file. Another problem, which may not be so obvious, occurs when a student takes more than the 4 courses which are provided for in the file. You could make a file with fields for 6 courses, but then you waste quite a bit of space in the file to allow for the few

students that take more than four courses. Another problem with creating a file with many fields occurs when you ask the computer to sort through the records to select certain ones. When large records are involved the computer has to work through the volumes of data that are not involved with the actual sorting operation you have asked it to accomplish.

Designing a good database requires an education which is beyond the scope of this tutorial. But a good rule of thumb to remember is that you should have many files of records with as few fields in each record. Minimize the number of fields - maximize the number of files.

One example of how to set up a database to be able to work with students who take different number of courses would be to set up one file with the students personal data such as name, age, address, student identification number, and major. Than set up second file with only two fields: student identification number and course number. This second file would have a separate record for each course the student takes. Now there is no problem when different students take a different number of courses. There is no space being held open for courses that are not taken as in the original file structure.

This is exactly what we will do. We will set up a file, course_stu, with two fields: student_id for student identification number and course_nu for course number.

In this section you will be working from the manual to build a database file on the computer. At this time go to the screen and read the paragraph explaining how to generate a database file, and then press any key to bring up the next screen.

Now on the screen you will see

"Enter the name of the new file:"

Type: COURSE_S (the character between the E and S is the underline character, formed by typing the uppercase "-" key just to the right of the zero key. The uppercase letters are formed by simultaneously pressing the shift key, located on the left and right side of the row above the space bar and the desired key for the uppercase letter)

Press: ENTER

You will now see the screen which will allow you to define the file structure as shown in Figure A.7.

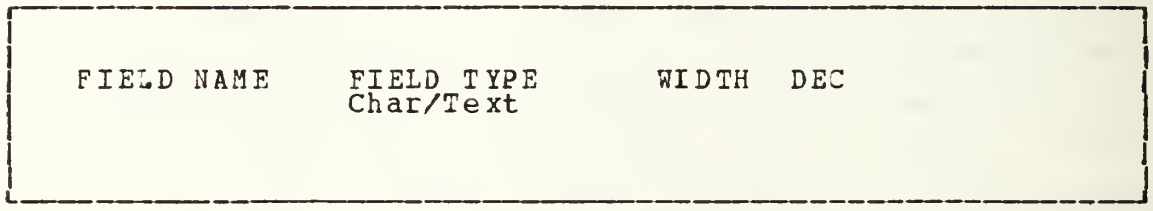

Figure A.7 Screen to Define a File Structure

The file structure you will enter into the computer is shown in Figure A.8. This structure will determine the makeup of the file. Detailed directions for entering the file structure are given below.

To enter the structure do the following:

TYPE:	ID_STU	(to name the first field)
PRESS:	ENTER	(the cursor will move to Field Type)
TYPE:	N	(to define the field as numeric)
PRESS:	ENTER	(the cursor will move to Width)
TYPE:	4	(to define a width of 4
PRESS:	ENTER	(the cursor will move to Dec field)
PRESS:	ENTER	(to define zero decimal places, the

Field Name	Field Type	Width	Dec
ID_STU	NUMERIC	4	
COURSE_NU	NUMERIC	4	

Figure A.8 File Structure

cursor will move to the next field name)

TYPE: COURSE_NU (to name the second field)

PRESS: ENTER (the cursor will move to numeric field)

TYPE: 4 (to define a width of 4)

PRESS: ENTER (the cursor will move to Dec)

PRESS: ENTER (to define zero decimal places, the cursor will move to the next field name)

PRESS: ENTER (to tell the system you have completed defining the file, the bottom of the screen will display the message:

"Hit Return key to confirm".

This is asking you to confirm that you have completed defining the fields of the file. Since we are finished defining the file structure, follow the instructions the microcomputer has given you:

PRESS: ENTER

At this point dBASE III will give you the option of entering records into your new file structure by displaying the message:

"Input records now? (y/n)

TYPE: "y" (for yes)

Table I provides the data you will enter into the file. Do not enter this data now. Detailed directions for entering the data are given at a later point in this tutorial. Each of these fields have a width of 4 digits. When you input the data into the fields the cursor will automatically jump to the next field upon being filled up. Since each student identification number (student_id) is 4 digits long, the cursor will jump to the course number field upon the entry of the fourth digit. Similarly, upon the entry of a fourth digit of a course number (course_nu), the cursor will jump to the next record, ready for data entry in the student_id field. When you have an entry that does not fill up a field width, such as will occur for course number 155, you will have to push the enter key after completing the entry.

TABLE I

Data for Student File

student_id	course_nu	student_id	course_nu
1000	2810	1101	3010
1050	3010	1399	3020
1399	3010	1222	3020
1222	155	1002	155
1002	2810	1000	155
1190	155	1190	3010
1171	3020	1171	3010
1157	3010	1050	3020
1150	2810	1157	155
1101	2810	1190	3010

We are now ready to enter the data into the records of the COURSE_S file. When entering the numbers use the numbers on the top row of the keyboard. Figure A.9 shows you what you will see on the screen.

Figure A.9 System Ready to Receive Input to Records

RECORD#	TYPE	THEN
1	1000	The cursor will beep and jump to the course_n field.
	2810	The cursor will beep and jump to the next record.
2	1050	
	3010	
3	1399	
	3010	
4	1222	
	155	Notice the cursor did not jump to the next record after the third digit was entered. Press the enter key to jump to the next record.
5	1002	
	2810	
6	1190	
	155	Press the enter key.
7	1171	
	3020	
8	1157	
	3010	
9	1150	
	2810	
10	1101	

```
            2810
11          1101
            3010
12          1399
            3020
13          1222
            3020
14          1002
            155     Press the enter key.
15          1000
            155     Press the enter key.
16          1190
            3010
17          1171
            3010
18          1050
            3020
19          1157
            155     Press the enter key.
20          1190
            3010    Push the enter key twice to stop
                    entering data into the file.
```

d. Edit Mode

The edit mode displays one record at a time on the screen for the user of the system to make either corrections to data entered incorrectly, or to make a change to an item such as an address change. While in the edit mode you can move back and forth between the records and fields by using the arrow keys on the numeric keypad.

To leave the edit mode there are several methods available. You can press the:

1. Ctrl key and End key simultaneously. Press the control key (Ctrl), located to the left of the "A"

key, and the End key, located on the "1" key of the numeric keypad simultaneously, and you will exit the edit mode saving all the changes entered during the current editing session.

2. Enter key. Pushing the enter key while the cursor is positioned at the last field of the last record will also save any changes made during the current session.
3. Esc key. The escape key (Esc), located to the left of the "1" key on the typewriter keyboard, will leave the edit mode without recording any changes made during the current editing session. This method is not recommended to be used.

Follow the instructions below when we enter the edit mode in just a moment, to change the course number from 3020 to 3010 in record number 7.

1. Advance the cursor to display record number 7 using the arrow keys.
2. Put the cursor down to the COURSE_NU field using the down arrow key.
3. Type in 3010.
4. Look at the rest of the records if you like, but be sure to exit the mode using the Ctrl and End keys as described above.

```
**********************************
*    GO TO THE SCREEN NOW        *
**********************************
```

e. Browse Mode

The browse mode displays as many records of a file as will fit on the screen. If there are more fields or records than would fit on the screen, you can use the arrow keys on the numeric keypad to scroll the screen to view them.

While viewing the records in the browse mode check the change you made while in the edit mode to record number 7. If it or any of the other records are incorrect make any change to put them in the proper form as shown Table I above.

To exit the browse mode there are two options. You can press the:

1. Ctrl and End keys simultaneously to save any changes made during the session.
2. Esc key to exit without recording any of the changes.
3. The enter key will not cause you to exit the browse mode.

When you are ready to leave the browse mode use one of the above methods to continue with the tutorial.

****
* GO TO THE SCREEN NOW *
****

f. Sorting

Sorting database files allows you to reorder a previously established file. Sorting a file causes the records to be physically rearranged in another file. The order of the rearranged file depends upon which field you chose to sort on, as you can sort the file on any field in the file. You may even select ascending or descending ordering. This is useful when putting records in alphabetical order (ascending), or ranking by grades (descending).

I'll show you an example by sorting the file "STUDENT".

****
* GO TO THE SCREEN NOW *
****

g. Indexing

Although sorting is a useful method of ordering a file, it also is very time consuming activity. There are two reasons for this. First, every time you sort a file the computer must determine the correct ordering and physically record a copy of the reordered file under a new file name. Secondly, every time you append or edit a record in the file you must give the sort command to have a sorted file again.

The computer has another method, known as indexing, to keep track of the ordering of records within a file. When a file ordering is generated by indexing, another file is created. This index file, when used with the database, will cause the records to appear in the order you directed when listed to the screen or a printer. The database, however, is still physically stored in the same order in which you entered the data into the file. The ordering is caused by an indexing file which keeps track of the ordering by storing pointers to the records. These pointers are stored in a manner that will cause the records to be displayed in the requested order.

Indexing a file has several advantages over sorting. The first is realized when you append a record or edit an open file. In this case the indexed file will note where the new record should be placed according to the ordering of the file. A second advantage is the time saved when a file is appended to or changed, because it is not necessary to resort the file to get the file back in the correct order. If you had multiple index files for a database file, such as one for ordering by the students name, city and major, each of the indexed files would be updated when a record is appended or edited.

One disadvantage with index files is that they can only be indexed in ascending order. This is a minor

annoyance when you consider the advantages. When you require a descending ordering you could have the database file sorted to a separate file.

****
* GO TO THE SCREEN NOW *
****

h. Appending To An Indexed File

Now you will append a record to the end of the file called STUDENT while in the index mode of operation. This will use the index file to remember where the new record should be alphabetically. The database file will still place it at the end of the file. Ensure you use uppercase letters. When you go to the screen use the instructions given below to append the file.

****
* GO TO THE SCREEN NOW *
****

TYPE	THEN
1219	cursor will beep and move to next field
ALBERTS	push enter
RALPH	push enter
K	cursor will beep and move to next field
PHY	cursor will beep and move to next field
JR	cursor will beep and move to next field
030164	cursor will beep and move to next field

Press enter to indicate that you have completed appending the file.

i. Join Command

As I mentioned previously, it is best to make many files with a few fields, rather than one file with many fields. This eliminates the need to manipulate a file with many fields each time you operate with the file. When you have limited the fields in your files there will be times when you will want to put together data which is held in separate files, such as the students names in a professors classes. With this data being held in the files STUDENT, COURSE_S, PROFESSO, and COURSE_P, it would be quite a task to manually track through these four files to retrieve the required data.

The join command will do this task for us. This command will join two files to a new file. You must specify the two files to be joined and the file name to which the results are to be stored. You must also specify the condition for which the compared fields will be joined. An example of this is when joining the files STUDENT and COURSE_S, you would stipulate that the files be joined when the student identification numbers, (ID_STU), in the two files match. The join command also lets you select which fields are to be recorded in the new file. You can select any number of fields from the two files which you are joining to be in the new file.

Sometimes the data you require must be retrieved from several files. Such is the case which I will demonstrate for you on the computer. There are, stored in the computer, two files you have not yet used, PROFESSO and COURSE_P. PROFESSO contains the professors identification number, name, department, and rank. COURSE_P contains the course number a professor is teaching and the professors identification number. This is similar to COURSE_S, which contains course number and student identification number.

```
*******************************
*    GO TO THE SCREEN NOW     *
* TO VIEW NEW FILE LISTINGS   *
*******************************
```

(1) First Join. To get a file which contains the course number, name of the instructor, and the students enrolled in the course requires three separate join commands.

The first join will be between the files PROFESSO and COURSE_P, which will cause a record to be created when the professors identification number is identical in the two records being compared. The new file will contain records with the fields that contain the professors names and course numbers being taught by the professors.

After each join is accomplished return to the tutorial before continuing to the next screen for information about what is to happen.

```
*********************************
*      GO TO THE SCREEN NOW     *
*      FOR THE FIRST JOIN       *
*********************************
```

(2) Second Join. The second join will be between the file created by the first join and COURSE_S, which will cause a record to be created when the course numbers are identical in the two records. The new file will contain records with the fields that contain the professors names, course numbers, and the identification numbers of the students taking the course.

```
*********************************
*      GO TO THE SCREEN NOW     *
*      FOR THE SECOND JOIN      *
*********************************
```

(3) Third Join. The third join will be between the file created by the second join and STUDENT, which will cause a record to be created when the student identification numbers are identical in the two records. The new file will contain records with the fields that contain the professors names, course numbers, and the names of the students taking the course.

```
*******************************
*     GO TO THE SCREEN NOW    *
*      FOR THE THIRD JOIN     *
*******************************
```

F. LEAVING THE TUTORIAL

When you desire to leave the tutorial select "Q" while in the main menu. If you are not at the main menu see paragraph A.4 above. When you have selected "Q" you will see what is called a prompt in the lower left corner which looks like - C>. This tells you that you are operating on drive C, which is the hard disk drive.

You are now ready to turn off the power. Do this by pushing in on the bottom of the power switch located in the power panel on the left side of the microcomputer. Once this is done unplug the plug from the wall socket, then unplug the black power cord from the surge suppressor. Now unplug the black power cord from the microcomputer. Coil up the black power cord and place it inside the power panel and slide the door back to close the power panel.

Remove the floppy disk from the disk drive. Do this by moving the horizontal lever on the front of the drive up and to the left (in a counterclockwise direction) to the vertical position. The floppy disk will pop out a slight bit. Remove the the disk from the drive and place it in the protection sleeve the instructor gave it to you in.

You are now ready to place the keyboard (or the base as it is soon to be) back on the microcomputer. Lift up the keyboard and place it over the screen and the disk drives with the keys on the inside. Now hold the base onto the microcomputer with one hand and with the other slide out the plastic latches on the left and right sides. Once the latches are locked out, flip in the two feet on the base. Now just one more item to complete securing the microcomputer. Lift the microcomputer up on to its base by the handle and flip in the two feet which elevated the screen.

See I told you that you would enjoy the course. I hope that this introduction to microcomputers has taken away some of the mystery they may have held in the past.

APPENDIX B

TUTORIAL SOFTWARE

```
**************************************************************************
*                                                                        *
*   FILE NAME:          START.PRG                                        *
*                                                                        *
*   AUTHOR:             LCDR S.T. COWEN III                              *
*                                                                        *
*   DATE CREATED:       1-04-85                                          *
*                                                                        *
*   CALLED BY:          BEGIN.PRG                                        *
*                                                                        *
*   CALLS:              INTRO.PRG                                        *
*                       INSTRU.PRG                                       *
*                       MAIN.PRG                                         *
*                                                                        *
*   VARIABLES USED:                                                      *
*                                                                        *
*   PRG FILES USED:                  DBF FILES USED:                     *
*   INTRO                                                                *
*   MAIN                                                                 *
*   QUIT                                                                 *
*   SUMMARY:  THIS MODULE STARTS THE TUTORIAL. IT CALLS THE              *
*             INTRODUCTION, INSTRUCTION, MAIN AND THE                    *
*             QUIT PROGRAM FILES.                                        *
*                                                                        *
*                                                                        *
**************************************************************************


clear
*THE FOLLOWING DISPLAYS THE PROGROM LOGO

@  4,36 SAY ' WELCOME '
@  6,37 SAY ' TO THE '
@  8,23 SAY ' PRACTICAL COMPTROLLERSHIP COURSE '
@ 10,28 SAY ' MICROCOMPUTER TUTORIAL '
@ 12,37 SAY ' AND '
@ 14,32 SAY ' BUDGET ANALYSIS '
@ 21,2 SAY '   '

WAIT
CLEAR
*THE FOLLOWING CALLS THE PROGRAM FILES WHICH RUN THE TUTROIAL


DO MAIN



*FINISHES THE PROGRAM

RETURN
```

```
*************************************************************************
*                                                                       *
*   FILE NAME:          MAIN.PRG                                        *
*                                                                       *
*   AUTHOR:             LCDR S.T. COWEN III                             *
*                                                                       *
*   DATE CREATED:       12-10-84                                        *
*                                                                       *
*   CALLED BY:          START.PRG                                       *
*                                                                       *
*   CALLS:              SPREAD.PRG                                      *
*                       DBASE,DBASE2,DBASE3,DBASE4                      *
*                       BUDGET                                          *
*                                                                       *
*   VARIABLES USED:     CHOICE                                          *
*                                                                       *
*   PRG FILES USED:                   DBF FILES USED:                   *
*       SPREAD                                                          *
*       DBASE,DBASE2,DBASE3                                             *
*       DBASE4,BUDGET                                                   *
*   SUMMARY:  DISPLAYS THE MAIN MENU, WITH SELECTIONS OF SPREADSHEET    *
*             OR DATABASE TUTORIALS, BUDGET ANALYSIS, OR EXIT           *
*             FROM THE PROGRAM.                                         *
*                                                                       *
*                                                                       *
*************************************************************************


CLEAR
*SET UP TO DISPLAY THE MAIN MENU

CLOSE DATABASES
SET TALK OFF
*DEFINE CHOICE AS THE VARIABLE TO WHICH THE USERS SELECTION IS STORED

CHOICE = 'X'
DO WHILE CHOICE <> 'Q'
    @ 2,31 SAY 'TUTORIAL SELECTION'
    @ 5,20 SAY '1 => SPREADSHEET TUTORIAL'
    @ 7,20 SAY '2 => DATABASE TUTORIAL'
    @ 9,20 SAY '3 => BUDGET ANALYSIS'
    @11,20 SAY 'Q => QUIT'
    ?
    ?
    ?
    ?
    ?
    ?
    ACCEPT '          CHOOSE  1,2,3, OR Q ' TO CHOICE
```

```
    CLEAR
    DO CASE
         CASE CHOICE = '1'
         *SELECTED SPREADSHEET TUTORIAL
              DO SPREAD
         CASE CHOICE = '2'
         *SELECTED DATABASE TUTORIAL
         *PRINT TEXT
              TEXT


           **RETURN TO MANUAL FOR INTRODUCTION TO DATABASES**


              ENDTEXT
              WAIT
              *NEXT FOUR STEPS RUN THE DATABASE TUTORIAL
              DO DBASE
              DO DBASE2
              DO DBASE3
              DO DBASE4
              CLOSE DATABASES
              ERASE TA.DBF
              ERASE TB.DBF
              ERASE TE.DBF
              ERASE LAST.NDX
              ERASE TEMP1.DBF
              ERASE COURSE_S.DBF
              USE STUDENT
              DELETE RECORD 11
              CLOSE DATABASES
         CASE CHOICE = '3'
         *SELECTED BUDGET ANALYSIS
              DO BUDGET
         CASE CHOICE = 'Q'
         *SELECTED QUIT
              RETURN
    ENDCASE
ENDDO
*RETURN TO START.PRG TO QUIT THE TUTORIAL

RETURN
```

```
*************************************************************************
*                                                                       *
*   FILE NAME:          SPREAD.PRG                                      *
*                                                                       *
*   AUTHOR:             LCDR S.T. COWEN III                             *
*                                                                       *
*   DATE CREATED:       12-10-84                                        *
*                                                                       *
*   CALLED BY:          MAIN.PRG                                        *
*                                                                       *
*   CALLS:              LOTUS SPREADSHEET USING RUN COMMAND             *
*                                                                       *
*   VARIABLES USED:                                                     *
*                                                                       *
*   DBF FILES US.                       WKS FILES USED:                 *
*                                       BUDGETX                         *
*                                                                       *
*                                                                       *
*   SUMMARY:  THIS IS THE BEGINNING OF THE SPREADSHEET TUTORIAL.        *
*             CALLS UP LOTUS 1-2-3 TUTORIAL AND SPREADSHEET             *
*                                                                       *
*                                                                       *
*********    ********    ***********************************************

CLEAR
*PRINT TEXT

TEXT

              **RETURN TO MANUAL FOR INTRODUCTION TO SPREADSHEETS**


ENDTEXT
WAIT
CLEAR
*PRINT TEXT

TEXT

    The system will now load the LOTUS 1-2-3 Spreadsheet Tutorial

into the computer for you to use.


ENDTEXT

WAIT
*LEAVES DBASE III AND CALLS UP LOTUS TUTORIAL

RUN TUTOR
*RETURNS FROM LOTUS TUTORIAL TO DBASE III
```

```
CLEAR
*PRINTS TEXT TO SCREEN

TEXT

      **RETURN TO MANUAL FOR INSTRUCTIONS ON COMPLETING A SPREADSHEEET**

ENDTEXT

WAIT
CLEAR
*LEAVES DBASE III AND CALLS UP LOTUS SPREADSHEET

RUN 123
*RETURNS  FROM LOTUS SPREADSHEET TO DBASE III

CLEAR
RETURN
```

```
**************************************************************************
*                                                                        *
*   FILE NAME:          BUDGET.PRG                                       *
*                                                                        *
*   AUTHOR:             LCDR S.T. COWEN III                              *
*                                                                        *
*   DATE CREATED:       12-10-84                                         *
*                                                                        *
*   CALLED BY:          MAIN.PRG                                         *
*                                                                        *
*   CALLS:              LOTUS 1-2-3                                      *
*                                                                        *
*   VARIABLES USED:                                                      *
*                                                                        *
*   DBF FILES USED:                  SCRATCH FILES USED:                 *
*                                    BUDGET4.WKS                         *
*                                                                        *
*                                                                        *
*   SUMMARY:  STARTS THE BUDGET ANALYSIS.                                *
*                                                                        *
*                                                                        *
*                                                                        *
**************************************************************************



CLEAR

*PRINTS TEXT
TEXT

            **RETURN TO MANUAL FOR BUDGET ANALYSIS INTRODUCTION**

ENDTEXT
WAIT
CLEAR

*CALLS LOTUS 1-2-3 TO USE IN BUDGET ANALYSIS

RUN 123


*RETURN TO MAIN.PRG
RETURN
```

```
*************************************************************************
*                                                                       *
*   FILE NAME:          DBASE.PRG                                       *
*                                                                       *
*   AUTHOR:  LCDR S.T. COWEN III                                        *
*                                                                       *
*   DATE CREATED: 12-10-84                                              *
*                                                                       *
*   CALLED BY:  MAIN.PRG                                                *
*                                                                       *
*   CALLS:                                                              *
*                                                                       *
*   VARIABLES USED:                                                     *
*                                                                       *
*   PRG FILES USED:       DBF FILES USED:                               *
*                         STUDENT                                       *
*                                                                       *
*                                                                       *
*   SUMMARY:  STARTS THE DATABASE TURORIAL.                             *
*                                                                       *
*                                                                       *
*                                                                       *
*************************************************************************
*


CLEAR
*PRINT TEXT

*FRAME1111111111111111111111
*PRINT TEXT

TEXT

    The first item we will look at will be how a database file looks.  There
are different ways in which a file can be viewed.  Some allow you to make
modifications to the file while viewing the data and others do not.  Lets take
a look at a file called "STUDENT", which is a file of students ID numbers,
names, majors, year, and birth date.


ENDTEXT
*XXXXXXXXXXXXXXXXXXXXXXXXXXXX

WAIT
CLEAR
```

```
*FRAME22222222222222222222

* OPEN DEMO FILE (STUDENT)
USE STUDENT

* DISPLAY DEMO FILE IN RAW FORM
LIST
*PRINT TEXT

TEXT
-------------------------------------------------------------------------------

    This file called "STUDENT" was opened by th  ommand "USE STUDENT" It was
then put to the screen by the command "LIST".      it in both of these
instructions for you.
    Notice the column headings.  These are known as field names in a database.
The field names relate an attribute to the subject record.
    Next we shall look at what is called the structure of the database file.
This will show you what the defined limits have been set to for the file.

ENDTEXT
WAIT
CLEAR
TEXT


          **RETURN TO MANUAL FOR INTRODUCTION TO FILE STRUCTURES**


ENDTEXT
*XXXXXXXXXXXXXXXXXXXXXXXXXXXX

WAIT
CLEAR
*FRAME333333333333333333333333
*DISPLAY THE DEMO FILE (STUDENT) STRUCTURE

DIS  AY STRUCTURE
*PF  T TEXT

TEXT
-------------------------------------------------------------------------------
    Notice the information displayed when the structure of a file is displayed.
In the top portion you are shown the name of the file (STUDENT), the number of
records currently in the file (10), and the date the file was last updated.
The lower portion displays information about the individual fields: The field
numbers (1 thru 7), the field names (ID_STU, L_NAME, F_NAME, MID_INT, MAJOR,
YEAR, and BIRTH_DATE ), the field type, the width of each field, and the total
width of a single record.
ENDTEXT
*XXXXXXXXXXXXXXXXXXXXXXXXXXXX
WAIT
CLEAR
```

```
*FRAME4444444444444444444444
*SHOW STRUCTURE OF DEMO FILE (STUDENT) AGAIN

DISPLAY STRUCTURE


*PRINT TEXT

TEXT
-----------------------------------------------------------------------
-------- The file name, number of fields, field name, type of field,
width and number of decimal places are determined by the user of the
system.
    The types of fields allowable are character (C), numeric (N),
logical (L), date (D), and memo (M). A character field can hold any
letter, number or symbol.  A numeric field can hold only numbers, a
plus or minus sign, and a decimal point.  A logical field only takes
one space to indicate true or false.  A date field will contain eight
characters to store a date as MM/DD/YR.  A memo field can contain text
up to 4000 characters in length.
ENDTEXT
*XXXXXXXXXXXXXXXXXXXXXXXXXXXX

WAIT
CLEAR
*RETURN TO MAIN.PRG TO GO TO DBASE3.PRG

RETURN
```

```
*************************************************************************
*                                                                       *
*   FILE NAME:          DBASE2.PRG                                      *
*                                                                       *
*   AUTHOR:             LCDR S.T. COWEN III                             *
*                                                                       *
*   DATE CREATED:       1-25-85                                         *
*                                                                       *
*   CALLED BY:          MAIN.PRG                                        *
*                                                                       *
*   CALLS:                                                              *
*                                                                       *
*   VARIABLES USED:                                                     *
*                                                                       *
*   PRG FILES USED:                     DBF FILES USED:                 *
*                                                                       *
*                                           COURSE_S                    *
*                                                                       *
*   SUMMARY:  CONTINUES THE DATABASE TUTORIAL, FRAMES 5 - 12            *
*             CREATE DATABASE FILE, EDIT AND BROWSE FUNCTIONS           *
*                                                                       *
*                                                                       *
*************************************************************************

CLEAR
*FRAME555555555555555555555555
*PRINT TEXT

TEXT
              **RETURN TO THE MANUAL FOR INTRODUCTION**
                   **TO CREATING A DATABASE FILE**


ENDTEXT
WAIT
CLEAR
TEXT

    Now you will generate a database file. I will enter the command "CREATE" to
start you off.  Push the enter keys between entries into the database to move
the coursor to the next field, and twice when you have completed all the
entries.  You can use the up/down/right/left arrow keys on the numeric key pad
on the right side of the keyboard to move the coursor back to any field you
want to change before you have pushed the enter key twice in a row.  Select "Y"
for yes when asked if you want to make record entries now.
ENDTEXT
*ENDXXXXXXXXXXXXXXXXXXXXXXXXXXX
WAIT
CLEAR
*FRAMES6666666677777777777777
* STUDENT CREATE A DATABASE FILE (COURSE_S)

CREATE
```

```
*ENDXXXXXXXXXXXXXXXXXXXXXXXXXXX
WAIT
CLEAR


*FRAME88888888888888888888888888
*PRINT TEXT

TEXT
    Now that you have built a database file you will want to check it for
accuracy.  There are a couple of ways to do this.  The first method we will
look at is called the edit mode.  This mode shows you one record at a time,
allowing you to move around using the four arrow keys, and the enter key when
you make a change.

ENDTEXT
WAIT
CLEAR
TEXT

            **RETURN TO THE MANUAL FOR INTRODUCTION TO EDIT MODE**


ENDTEXT
*ENDXXXXXXXXXXXXXXXXXXXXXXXXXXX

WAIT
CLEAR
*FRAME99999999999999999999999999
*STUDENT EDIT DEMO FILE (COURSE_S)

EDIT 1
*ENDXXXXXXXXXXXXXXXXXXXXXXXXXXX

CLEAR

*FRAME-10-10-10-10-10-10-10-10
*PRINT TEXT

TEXT
    We shall now look at a mode which allows you to scan and revise
the complete file on the screen at the same time, or at least as much
as will fit on the screen at one time.  This mode is the BROWSE mode.
You can move around the screen with the four arrow keys.  To exit this
mode press the "CTRL" key to the left of the "A" key and the "END" key
on the numeric key pad simultaneously.  This will save any changes
made while browsing.  If you do not want to save the changes, the
"ESC" key to the left of the "1" key on the left side of the key board
will leave the browse mode without saving any changes made during the
session.

ENDTEXT
```

```
CLEAR

TEXT
          **RETURN TO THE MANUAL FOR INTRODUCTION TO BROWSE MODE**


ENDTEXT
*ENDXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
WAIT

*ENSURE THE RECORD POINTER IS AT RECORD #1 OF DEMO FILE (COURSE_S)
*STUDENT BROWSE THE FILE

GOTO 1
*FRAME-11-11-11-11-11-11-11-11-11

BROWSE
*ENDXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

*FRAME-12-12-12-12-12-12-12-12-12
*PRINT TEXT

TEXT
    A feature of the browse mode is the capability to limit the
viewing fields to the ones that are of interest.  This is accomplished
by telling the computer what fields you want to view after the browse
command, as indicated below.

              "BROWSE FIELDS XXXXXX,YYYYYY,ZZZZZZ

Where "XXXXXX,YYYYYY,ZZZZZZ" are the fields you want displayed on the screen.
This is useful if you are reviewing a file with many fields.  Remember the
first file I showed you with data about students.  It had many fields of
information.  Lets take a look at it again.
ENDTEXT
*ENDXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
WAIT
CLEAR
*RETURN TO MAIN.PRG TO GO TO DBASE3.PRG

RETURN
```

```
*************************************************************************
*                                                                       *
*   FILE NAME:        DBASE3.PRG                                        *
*                                                                       *
*   AUTHOR:           LCDR S.T. COWEN III                               *
*                                                                       *
*   DATE CREATED:     1-28-85                                           *
*                                                                       *
*   CALLED BY:        MAIN.PRG                                          *
*                                                                       *
*   CALLS:                                                            *
*                                                                       *
*   VARIABLES USED:                                                   *
*                                                                       *
*   PRG FILES USED:                  DBF FILES USED:                  *
*                                    STUDENT     LAST.NDX             *
*                                    TEMP1                            *
*                                                                       *
*   SUMMARY:  CONTINUES WITH THE DATABASE TUTORIAL.  FRAMES           *
*             14 - 21.  DISPLAYS BROWSE, SORT, AND INDEX COMMANDS.   *
*                                                                       *
*                                                                       *
*                                                                       *
*************************************************************************

*FRAME-14-14-14
*OPEN DEMO FILE (STUDENT) FOR LISTING

USE STUDENT
LIST
*PRINT TEXT

TEXT
-------------------------------------------------------------------------
    Now lets look at just the students last name, major, and year instead of
all the fields.  Remember to use the "CTRL'" and "END" keys simultaneously to
leave the browse mode
ENDTEXT
*ENDXXXXXXXXXXXX
WAIT

*FRAME-15-15-15
*PLACE DEMO FILE (STUDENT) RECORD POINTER TO RECORD #1
*THEN BROWSE FIELDS L_NAME, MAJOR, AND YEAR

GOTO 1
BROWSE FIELDS L_NAME,MAJOR,YEAR
*ENDXXXXXXXXXXXX

WAIT
CLEAR
```

```
*FRAME-16-16-16
*PRINT TEXT

TEXT

                **RETURN TO MANUAL FOR INTRODUCTION TO SORTING**

ENDTEXT
WAIT
CLEAR
TEXT


    Now that you have constructed a database file and made changes, lets do
something with the file that shows you the power of a database.  The SORT
command allows you to sort a file on any of the fields.  You can put the file
in alphebetical order, sorted on the students last name, first name, major, or
any of the fields you select.  You can even sort in reverse alphebetical order.
    To do this the computer will create another database file.  That way the
file being sorted will stay in its original form.

    The command I will enter to sort the file "STUDENTS" is:

    USE STUDENTS     This opens the file to be sorted.

    SORT ON L_NAME TO TEMP1
                     This will sort the file on the students Last name, and
                     store the sorted file in a file called TEMP1.
ENDTEXT
*ENDXXXXXXXXX
WAIT
CLEAR
*OPENING TWO DEMO FILES (STUDENT AND TEMP1) FOR SORTING
*STUDENT SORTED ACCORDING TO L_NAME AND STORED IN TEMP1

SELECT 1
USE STUDENT
SORT ON L_NAME TO TEMP1
SELECT 2
USE TEMP1

*FRAME-17-17-17
LIST
*PRINT TEXT

TEXT
--------------------------------------------------------------------------------

    HOW ABOUT IF I SHOW YOU BOTH THE SORTED FILE AND
ORIGINAL FILE?  THAT WAY YOU CAN SEE THE DIFFERENCE.
ENDTEXT
*ENDXXXXXXXXXXXX
WAIT
CLEAR
```

```
*FRAME-18-18-18
*DISPLAY SORTED DEMO FILE (TEMP1) AND UNSORTED DEMO FILE (STUDENT)

LIST
SELECT 1
LIST
*ENDXXXXXXXXXXXX

WAIT
CLEAR
*FRAME-19-19-19
*PRINT TEXT

TEXT


          **RETURN TO MANUAL FOR INTRODUCTION TO INDEX COMMAND**


ENDTEXT
*ENDXXXXXXXXXXXX
WAIT
CLEAR
*USE DEMO FILE (STUDENT) TO BE INDEXED ON FIELD L_NAME TO DEMO FILE (LAST)
*THEN OPEN DEMO FILE (LAST) AND DISPLAY INDEXED FILE

SELECT 1
INDEX ON L_NAME TO LAST
SET INDEX TO LAST
*FRAME-20-20-20
LIST
*PRINT TEXT

TEXT
-------------------------------------------------------------------------------
    Notice the ordering of the records are alphabetical by last name.  But also
notice the ordering of the record numbers.  The records are still physically
filed in the order in which they were entered into the database file.

ENDTEXT
WAIT
CLEAR
TEXT


     **RETURN TO MANUAL FOR INSTRUCTIONS TO APPENDING THE INDEXED FILE**

ENDTEXT
```

```
WAIT
*APPEND A RECORD TO DEMO FILE (LAST) ACCORDIN TO INSTRUCTIONS IN MANUAL

APPEND
*FRAME-21-21-21
LIST
*PRINT TEXT

TEXT
-------------------------------------------------------------------------------
    Notice the record you have just added to the file is in the correct
alphabetical position.  Also notice the record number associated with this
record.  It is the last number, indicating that it is the last physical record
in the file.

ENDTEXT
*ENDXXXXXXXXXXX
WAIT
CLEAR

*OPEN DEMO FILE (STUDENT) TO DISPLAY HOW THE RECORD APPENDED TO AN
*INDEX FILE IS STORED PHYSICALLY AT THE END OF THE FILE

SELECT 1
USE STUDENT
LIST
TEXT

-------------------------------------------------------------------------------

    Notice that the record you added is at the end of the file.

ENDTEXT
WAIT
CLEAR

*RETURN TO MAIN.PRG TO GO TO DBASE4.PRG

RETURN
```

```
**************************************************************************
*                                                                        *
*   FILE NAME:        DBASE4.PRG                                         *
*                                                                        *
*   AUTHOR:           LCDR S.T. COWEN III                                *
*                                                                        *
*   DATE CREATED:     1-28-85                                            *
*                                                                        *
*   CALLED BY:        MAIN.PRG                                           *
*                                                                        *
*   CALLS:                                                               *
*                                                                        *
*   VARIABLES USED:                                                      *
*                                                                        *
*   PRG FILES USED:                  DBF FILES USED:                     *
*                                    PROFESSO, COURSE_P, STUDENT,        *
*                                    TA, TB, TE                          *
*                                                                        *
*   SUMMARY: CONTINUE DATABASE TUTORIAL.  FRAMES 22 - 28.                *
*            DEMONSTRATES JOIN COMMAND.                                  *
*                                                                        *
*                                                                        *
**************************************************************************

*FRAME-22-22-22
*PRINT TEXT

TEXT

          **RETURN TO MANUAL FOR INTRODUCTION TO JOIN COMMAND**

ENDTEXT
*ENDXXXXXXXXX
WAIT
CLEAR

*FRAME-23-23-23
*OPEN DEMO FILE (PROFESSO) AND DISPLAYS

USE PROFESSO
LIST
*PRINT TEXT

TEXT

--------------------------------------------------------------------------
    This is the listing of the database on the instructors we will be looking
at.  There is also a database which holds the courses they are teaching this
quarter.  Lets look at it now.
ENDTEXT
*ENDXXXXXXXXX
WAIT
CLEAR
```

```
*FRAME-24-24-24
*OPEN DEMO FILE (COURSE_P) AND DISPLAY

USE COURSE_P
LIST
*PRINT TEXT

TEXT
--------------------------------------------------------------------------------
    Can't really tell very much from this listing can you?  This is where the
real power of a database will show its true colors.  There is a command which
allows you to join records with similar fields from two separate database
files.  Lets take a look at the JOIN command.

ENDTEXT
WAIT
CLEAR
TEXT


          **RETURN TO MANUAL FOR INFORMATION ON THE FIRST JOIN**


ENDTEXT
*ENDXXXXXXXX
WAIT
CLEAR
*CLOSE DEMO FILES (PROFESSO AND COURSE_P) IN PREPARATION TO OPEN
*MULTIPLE DEMO FILES (PROFESSO AND COURSE_P) IN PREPARATION
*JOIN COMMAND

CLOSE DATABASES
SELECT 1
USE PROFESSO
SELECT 2
USE COURSE_P
JOIN WITH PROFESSO TO TE FOR ID_PROF = PROFESSO->ID_PROF;
 FIELDS COURSE_NU, PF_NAME, PL_NAME

*FRAME-25-25-25
*OPEN ANOTHER DEMO FILE (TE) TO DISPLAY THE RESULTS OF JOIN COMMAND

SELECT 4
USE TE
LIST
*PRINT TEXT

TEXT
--------------------------------------------------------------------------------
    Now you know what course number each professor is teaching.  How about the
students in each of the professors classes?
ENDTEXT
*ENDXXXXXXXXXX
WAIT
```

```
*OPEN ANOTHER DEMO FILE (COURSE_S) TO BE JOINED WITH DEMO FILE (TE)

SELECT 3
USE COURSE_S
SELECT TE
JOIN WITH COURSE_S TO TA FOR COURSE_NU = COURSE_S->COURSE_NU

*FRAME-26-26-26
*OPEN DEMO FILE (TA) TO DISPLAY RESULTS OF JOIN

SELECT 5
USE TA
LIST
*ENDXXXXXXXXXX
WAIT
CLEAR
*OPEN DEMO FILE (STUDENT) TO BE JOINED WITH DEMO FILE (TA)

SELECT 6
USE STUDENT
SELECT 5
USE TA
JOIN WITH STUDENT TO TB FOR ID_STU = STUDENT->ID_STU;
 FIELDS PF_NAME,PL_NAME, COURSE_NU,  F_NAME, L_NAME
*FRAME-27-27-27
*OPEN DEMO FILE (TB) TO DISPLAY RESULTS OF JOIN

SELECT 7
USE TB
LIST
*ENDXXXXXXXX
WAIT
CLEAR
*FRAME-28-28-28
*PRINT TEXT

TEXT
    In an actual application of this example the three joins would have been
included in a single program, and the final listing would have been the only
one which would have been saved and printed to the screen.
```

SUMMARY

The database portion of the tutorial is now completed. In this section you have been intorduced to database files, structures, creation, editing, browsing, sorting, indexing, and joining. This has not been intended to make you an expert in database manipulation, but rather give you a brief introduction into the world of microcomputer databases.

```
ENDTEXT
*ENDXXXXXXXXXXXX
WAIT
CLEAR
*RETURN TO MAIN.PRG TO DISPLAY MENU
RETURN
```

LIST OF REFERENCES

1. Wilcox R. J., "Computers On The Rise", Government Computer News, v. 3 , November 1984.

2. Johnson, B., "MIS Budgets Trail Workload", Government Computer News, v. 3, November 1984.

3. Heffernan, H., "AF Enthusiasm Shows Government Micro Surge" Government Computer News, v. 3, December 1984.

4. Judge, J. F., "GSA's Computer Store: The Concept That Works"' Government Executive, v. 16, July-August 1984.

5. Godfrey, D and Streling, S. The Elements of CAL, Reston, 1982.

6. Joint Educational Management Research, Report ED 211 062, A Standards Guide for the Authoring of Instructional Software, Crawford, S., 1981.

7. Barnes, K., "Computer-Based Training Commandments: Read and (Maybe) Sin No More" PC WEEK, v. 2, 19 February 1985.

8. Martin, C. L., "Best Selling SW" Personal Software, v. 2, December 1984.

BIBLIOGRAPHY

Fitzgerald, E.B., Office Integration-Vision vs. Reality, Computer and People, vol 33 no 1-2, Jan-Feb 84

Kroenke, D., Database Processing: Fundamentals, Design, Implementation, Scientific Research Associates, Inc, 1983

Lias, E.J., Should Businesses Buy Microcomputers For Their Employees?, Computer and People, vol 33 no 3-4, Mar-Apr 84

Lockett, K. V. and O'Neil, M. E., Simulation of an Automated Information System on a Microcomputer, Master's Thesis, Naval Postgraduate School, March1984

Mack, R.L., Clayton, H.L., and Carroll, J.M., Learning To Use Word Processors: Problems and Prospects, ACM Transactions on Office Informations Systems, vol 1 no 3, July 1933

Milgram, E., Microprocessors: Their Management and Use in Low Cost Applications, ACCESS The Navy Data Automation Review, vol 2 no 2, Mar-Apr 1977

Myers, R. H., Grimm, D. H., and Shumway, S. G., A Microcomputer Abstract Representation of the Marine Corps' SABRS Financial Management System, Master's Thesis, Naval Postgraduate School, Dec 84

Naval Postgraduate School Report NPS55-81-019, Enhancing Technology Transfer of Computer Hardware and Software Architectures Using Human Factors in Initial Design , by G. K. Poock and R. J. Roland, 1981

Rockart, J.F. and Morton, M.S.S., Computers and the Learning Process in Higher Education , McGraw-Hill, 1975

Seidel, R.J., It's 1980: Do You Know Where Your Computer Is?, Human Resources Research Organization, 1980

Senn, J.A., Information Systems in Management, Wadsworth Publishing Co., 1982

Shaw, S., GSA Opens 2nd Micro Store, Government Computer News, v. 3, no 11, Oct 84

Slater, L.W., Low Cost Expandable Microcomputer Systems, v. 3, 1981, pages 8-14

Weisgerber, R.A., Trends, Issues, and Activities in Individualized Learning, American Institutes for Research, 1972

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station, Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
3.	Department Chairman, Code 54 Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943	1
4.	CDR Frank E. Royer, Code, Code 54Ro Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943	1
5.	Dr. Norman Lyons, Code 54DH Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943	1
6.	LCDR Sherry T Cowen III Code 20 SOAC 85050 NAVSUBSCHOOL NAVSUBBASE BOX 700 Groton, CT 06349	2

13537 5

211187

Thesis
C756959 Cowen
c.1 A microcomputer tutorial on spreedsheets and databases with a simulated budget preparation.

thesC756959
A microcomputer tutorial on spreadsheets
3 2768 002 09005 2
DUDLEY KNOX LIBRARY